Weather Information Database (WIDB)
Information Technology System Architecture Document

Appendix C – Technical Standards for Consideration

THIS PAGE INTENTIONALLY LEFT BLANK.

NextGen Architecture and Infrastructure Development

WIDB IT System Architecture Document

Appendix C – Technical Standards for Consideration

Table of Contents

THIS PAGE INTENTIONALLY LEFT BLANK.

1 Introduction

This appendix provides a brief overview of numerous technical standards and protocols for interoperable exchange of data. This is presented as background information that covers a wide range of data standards, many of which will be taken into consideration for use in the 4-D Weather Cube.

1.1 Open Geospatial Consortium (OGC)

The Open Geospatial Consortium (OGC) is an international industry consortium of 347 companies, government agencies and universities participating in a consensus process to develop publicly available interface specifications. OpenGIS Specifications support interoperable solutions that "geo-enable" the Web, wireless and location-based services, and mainstream IT. The specifications empower technology developers to make complex spatial information and services accessible and useful with all kinds of applications.

OpenGIS® is a Registered Trademark of the Open Geospatial Consortium, Inc (OGC) and is the brand name associated with the Specifications and documents produced by the Open Geospatial Consortium, Inc (OGC). OpenGIS specifications are developed in a unique consensus process supported by OGC industry, government and academic members to enable geo-processing technologies to interoperate, or "plug and play".

Within the Web Services environment, OGC Web Services represent a mature, standards-based framework that enables seamless integration of a variety of online data. It allows different systems to communicate with each other across the Web using well established technologies such as XML and HTTP. OGC Web Services provide a vendor-neutral, interoperable framework for web-based discovery, access, integration, analysis, exploitation and visualization of multiple online geodata sources, sensor-derived information, and geo-processing capabilities.

The OGC Web Services concept is based on a model of a distributed service-oriented architecture operated by data stewards, which is in line with NOAA Enterprise Architecture principles.

1.1.1 Geography Markup Language (GML)

The Geography Markup Language (GML) is an OGC standard format to represent geo-spatial information with XML. The mechanisms and syntax that GML uses to encode spatial information in XML are defined in the specification of OpenGIS.

GML defines an XML schema for representing geographic features (attributes, geometry, relationships, coordinate reference systems, topology, time, units of measure, etc). GML establishes rich sets of advanced and complex features in representing the spatial data. GML is co-branded as ISO 19136, and is consistent with ISO 191xx family.

GML is capable of supporting any application schema, and numerous of them have been implemented, including, but not limited to, the following:

- Aeronautical Information Exchange Model (AIXM),
- Cyclone Warning Markup Language (CWML),
- Digital Weather GML (dwGML),
- GeoscienceMarkup Language (GeoSciML),
- Keyhole Markup Language (KML),
- City Geography Markup Language (CityGML),

1.1.1.1 GML Profiles

The GML specification declares a large number of XML elements and attributes meant to support a wide variety of capabilities. For example, the GML specification can encode dynamic features, spatial and temporal topology, complex geometric property types and coverages. With such a wide scope, interoperability can only be achieved by defining profiles of GML that deal with a restricted subset of GML capabilities, e.g. Point Profile, Simple Features Profile, GeoRSS. Such profiles limit the number of GML object types that can appear in compliant schemas and consequently are easier to process

Ref: http://www.opengeospatial.org/standards/gml

1.1.2 Keyhole Markup Language (KML)

Google submitted KML specification to the Open Geospatial Consortium to be evolved within the OGC consensus process. KML is an XML language focused on geographic visualization, which includes not only the presentation of graphical data on the globe, but also the control of the user's navigation in the sense of where to go and where to look. Google extensively uses data in KML format in the visualization tools like Google Earth and Google Ocean.

That said, KML complements such key existing OGC standards as GML, WFS and WMS. For example, some KML geometry elements (point, line string, polygon) have been derived from GML. OGC and Google have agreed that the future development of KML and GML can be synchronized in terms of use of the same geometry representation.

Ref: http://www.opengeospatial.org/standards/kml

1.1.3 Climate Science Modeling Language (CSML)

Strictly speaking, Climate Science Modeling Language (CSML) is not an OGC standard but due to its close relation to the OGC data model it is logically consistent to describe it within the OGC section.

The CSML is a standard-based data model and GML application schema for atmospheric and oceanographic data. It represents the logical structure and semantic content of a dataset. Specific data object types are called ‘feature types’. They provide a common information model for oceanographic and atmospheric data types. A dual purpose of CSML is to provide a ‘wrapper’ mechanism to encapsulate legacy file based data, casting them onto feature instances. The CSML work is being performed as part of the NERC DataGRID project.

The CSML attempts to encapsulate important semantics of climate science data in a generic manner. In essence, it provides an abstract semantic model for representing a range of data objects of relevance to climate science. Because of that, it may be used to build intelligent services for data sub-setting, aggregation, and processing.

Ref: http://ndg.nerc.ac.uk/csml

1.1.4 Web Map Service (WMS)

The OGC WMS produces maps of spatially referenced data dynamically from geographic information. This standard defines a "map" as a digital image file suitable for display on a computer screen (map is not the data itself). WMS-produced maps are generally rendered in a pictorial format such as PNG, GIF or JPEG, or occasionally as vector-based graphical elements in Scalable Vector Graphics (SVG) or Web Computer Graphics Metafile (WebCGM) formats.

WMS defines three operations: GetCapabilities provides a description of the shared information content and acceptable request parameters; GetMap returns a map as an image; and optional GetFeatureInfo provides information about features at a point on a map.

A standard web browser can submit requests to WMS in the form of Uniform Resource Locators (URLs). The content of such URLs depends on which operation is requested. In particular, when requesting a map the URL indicates what information is to be shown on the map, what portion of the Earth is to be mapped, the desired coordinate reference system, and the output image width and height. When two or more maps are produced with the same geographic parameters and output size, the results can be accurately overlaid to produce a composite map. The use of image formats that support transparent backgrounds (e.g., GIF or PNG) allows underlying maps to be visible. Furthermore, individual maps can be requested from different servers. The Web Map Service thus enables the creation of a network of distributed map servers from which clients can build customized maps. This International Standard applies to a Web Map Service that publishes its ability to produce maps rather than its ability to access specific data holdings. A basic WMS classifies its geographic information holdings into "Layers" and offers a finite number of predefined "Styles" in which to display those layers. This International Standard supports only named Layers and Styles, and does not include a mechanism for user defined symbolization of feature data.

Ref: http://www.opengeospatial.org/standards/wms/

1.1.5 Web Map Tile Service (WMTS)

The Web Map Tile Service (WMTS) is the OGC's proposed standard for tile-based web mapping - an open alternative to proprietary web mapping services like Google Maps. The candidate WMTS standard is much like OGC's WMS, but it enables faster server performance. It is an evolution of Tile Map Service Specification by OSGeo and the Tiled WMS by NASA OnEarth.

OGC Web Map Server (WMS) creates a new image for each request. That approach allows WMS to reach high level of interoperability: a WMS client can overlay map layers from many sources in an arbitrary bounding box at an arbitrary scale with any number of styles. However, such a WMS server is required to generate each requested map image on the fly it is slow to respond.

To improve performance, instead of creating a new image for each request, a WMTS returns small pre-generated images (e.g., PNG or JPEG) or reuses identical previous requests that follow a set of tile matrices; that allows almost "instant" zoom and pan.

The proposed standard provides support for multiple architectural patterns - KVP, REST and SOAP.

Ref: http://www.opengeospatial.org/pressroom/pressreleases/965

1.1.6 Web Feature Service (WFS)

Web Feature Service (WFS) defines protocols for requesting and (optionally) inserting and modifying GML Features through HTTP requests. GML Features are geospatial objects such as images or GML documents that describe real-world entities, e.g. meteorological objects (fronts, precipitation areas), weather observation readings, observation stations or roads as a combination of the entities' geometry (points, lines, or polygons) and attributes (other information about those entities).

According to the OGC's Reference Model:[1]

> A **feature** is an abstraction of a real world phenomenon. A **geographic feature** is a feature associated with a location relative to the Earth. A digital representation of the real world can be thought of as a set of features.[...] A feature is not defined in terms of a single geometry, but rather as a conceptually meaningful object within a particular information or application community, one or more of the feature's properties may be geometric.

The WFS specification allows a client to retrieve geospatial data encoded in GML from multiple WFS servers in a similar fashion as WMS allows a client to overlay map images for display served from multiple services on the Internet or through other net enable services. The WFS specification defines three basic operations: *GetCapabilities* queries the WFS service to determine available options; *DescribeFeatureType* retrieves the XML schema to allow the WFS client to parse the result; and *GetFeature* performs the actual query.

Ref: http://www.opengeospatial.org/standards/wfs/

1.1.7 Web Coverage Service (WCS)

The Web Coverage Service (WCS) is similar to WMS and WFS, but instead of images or features deals with coverages, i.e. digital geospatial information representing space-varying phenomena. A coverage is a feature that has multiple values for each attribute type, where each direct position within the geometric representation of the feature has a single value for each attribute type. As such, WCS operates with gridded data, i.e. collection of mappings from geometry (grid points) into parameter (temperature, pressure, etc.) values. WCS provides access to intact (unrendered) geospatial information, as needed for client-side rendering, multi-valued coverages, and input into scientific models and other clients beyond simple viewers.

[1] Open Geospatial Consortium Inc, "OGC Reference Model (ORM)", OGC 08-062r4, Version 2.0, November 11, 2008

Similar to WFS, the WCS client can request the WCS server for available data using *GetCapabilities* function; ask the server to describe the interesting coverages with *DescribeCoverage* function; and create and return the requested subset of the available coverage data with *GetCoverage* function.

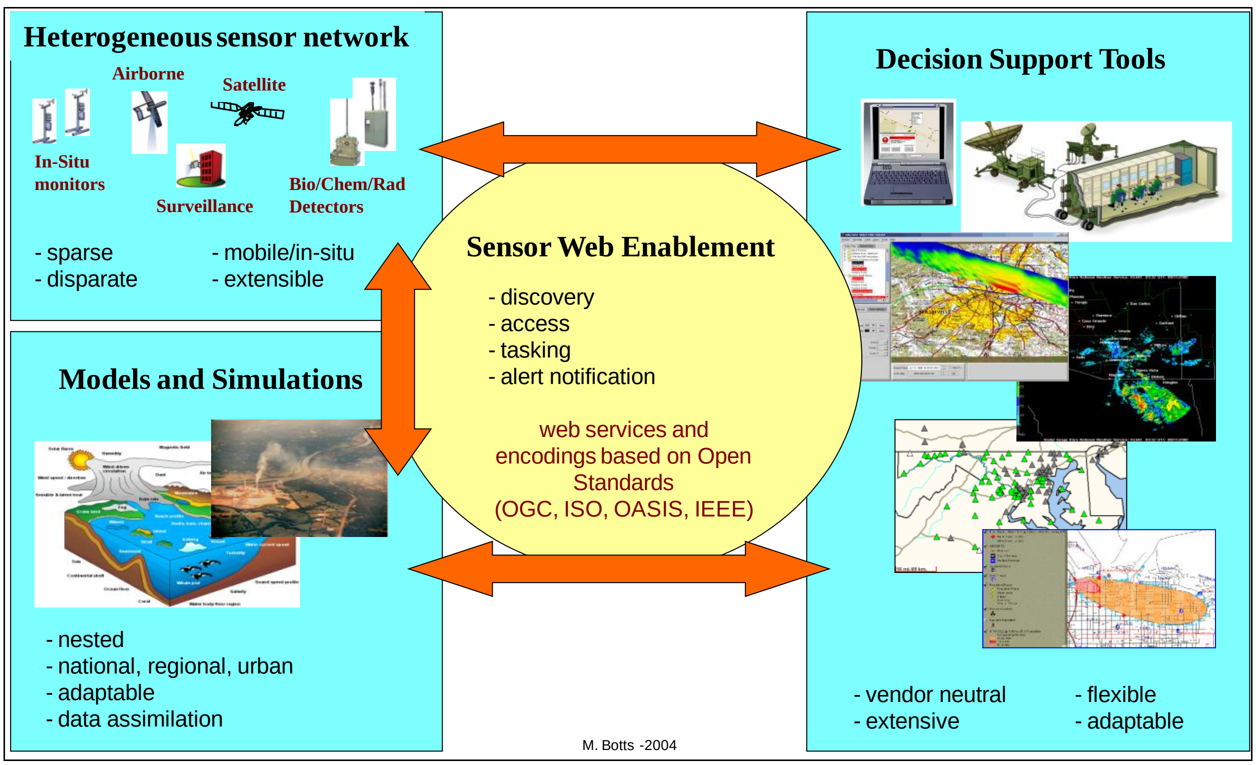
WCS is explicitly called out in the Global Earth Observation System of Systems (GEOSS) architecture and is supported by some commercial off-the-shelf (COTS) Geographic Information System (GIS) tools. WCS is also included into open-source Thematic Real-time Environmental Distributed Data Services (THREDDS) Java application developed by Unidata (see also OPeNDAP).

Ref: http://www.opengeospatial.org/standards/wcs/

1.1.8 Sensor Web Enablement

OGC Sensor Web Enablement (SWE) initiative is a framework of open standards for exploiting Web-connected sensors and sensor systems of all types. SWE is a Service Oriented Architecture (SOA) approach: its distributed architecture allows independent development of services while enabling interoperability and on-the-fly connectivity between resources. The SWE framework is illustrated in figure below:

Sensor Web Enablement Framework



The SWE activities provide specifications, interfaces and protocols to allow various applications and services to access, and control sensors over networks such as the Internet with the help of the Web standard technologies and protocols. The functionality that OCG has targeted within a sensor web includes:

- Discovery of sensor systems, observations, and observation processes that meet an application's or user's immediate needs;
- Determination of a sensor's capabilities and quality of measurements;
- Access to sensor parameters that automatically allow software to process and geo-locate observations;
- Retrieval of real-time or time-series observations and coverages in standard encodings
- Tasking of sensors to acquire observations of interest;
- Subscription to and publishing of alerts to be issued by sensors or sensor services based upon certain criteria.

OGC SWE standards include the following main OpenGIS® specifications:

- Observations & Measurements Schema (O&M) – standard models and XML Schema for encoding observations and measurements from a sensor, both archived and real-time.
- Sensor Model Language (SensorML) – standard models and XML Schema for describing sensors systems and processes.
- Transducer Markup Language (TransducerML or TML) – the conceptual model and XML Schema for describing transducers and supporting real-time streaming of data to and from sensor systems.
- Sensor Observations Service (SOS) - standard web service interface for requesting, filtering, and retrieving observations and sensor system information.
- Sensor Planning Service (SPS) – standard web service interface for requesting user-driven acquisitions and observations.

1.1.8.1 Observations and Measurements Schema (O&M)

O&M establishes a high-level framework for representing observations, measurements, procedures and metadata of sensor systems. O&M provides a standard model for representing and exchanging observation results, alleviating the need to support a wide range of sensor-specific and community-specific data formats.

The O&M specification defines observation as an event with a resulting value describing some phenomenon. An observation uses a procedure to determine the value, which may involve a sensor or observer, analytical procedure, simulation or other numerical processes.

Ref: http://www.opengeospatial.org/standards/o%2526m

1.1.8.2 Sensor Modeling Language (SensorML)

Sensor Model Language (SensorML) is a standard data model and GML application schema for describing the processes within sensor and observation processing systems as well as for modeling sensor metadata. SensorML provides an information model and encodings that enable discovery and tasking of Web-resident sensors, and exploitation of sensor observations.

SensorML provides a functional model of the sensor system, rather than a detailed description of its hardware. SensorML treats sensor systems and a system's components (e.g. sensors, actuators, and platforms) as processes; all processes, including sensors and sensor systems, have input, output, parameters, and methods that can be utilized by applications for exploiting observations from any sensor system.

SensorML has initially started in NASA and CEOS (Committee for Earth Observation Satellites) projects, and was brought into OGC later.

Ref: http://www.opengeospatial.org/standards/sensorml

1.1.8.3 Transducer Markup Language (TransducerML or TML)

A transducer is a superset of sensors and actuators. The Transducer Markup Language (TML) provides a method and message format for describing information about transducers and transducer systems. TML also provides a mechanism to capture, transport and archive transducer data, in a common form, regardless of the original source. TML utilizes XML for the capture and exchange of data.

The main purpose of TML is to provide means for compensation of sensor distortions. Sensor data is often an artifact of the sensor's internal processing rather than a true record of phenomena state. The effects of this processing on sensed phenomena are hardware-based and can be characterized as functions. TML goal is to model the known hardware behaviors, and to use the models to reverse distorting effects and return artifact values to the phenomena realm. TML provides models for a transducer's latency and integration times, noise figure, spatial and temporal geometries, frequency response, steady-state response and impulse response.

TML complements and has been harmonized with SensorML and O&M: TML provides an encoding and a conceptual model for streaming real-time "clusters" of time-tagged and sensor-referenced observations from a sensor system whereas SensorML describes the system models that allow a client to discover, interpret, and process the streaming observations.

Ref: http://www.opengeospatial.org/standards/tml

1.1.8.4 Sensor Observation Service (SOS)

Sensor Observation Service (SOS) is another part of SWE group of specifications. Used in conjunction with other OGC specifications, the SOS provides a broad range of interoperable capability for discovering, binding to and interrogating individual sensors, sensor platforms, or networked constellations of sensors in real-time, archived or simulated environments.

The SOS provides an API for managing deployed sensors and retrieving sensor data either from in-situ sensors (e.g., water monitoring) or remote sensors (e.g., satellite imaging). Used in conjunction with other OGC specifications, the SOS provides a broad range of interoperable capability for discovering, binding to and interrogating individual sensors, sensor platforms, or networked constellations of sensors in real-time, archived or simulated environments.

Similar to the other OGC Web services, SOS operates with three mandatory core functions: *GetCapabilities* provides the means to access SOS service metadata; *GetObservation* provides access to sensor observations and measurement data via a spatio-temporal query that can be filtered by phenomena; *DescribeSensor* retrieves detailed information about the sensors and the processes generating those measurements.

GetObservation function returns an O&M compliant document that includes procedure, a feature, properties of the feature being estimated and a result. The O&M specification leaves open the encoding format of the result. However, at several OGC projects, SweCommon is the encoding of preference.

Used in conjunction with other OGC specifications, the SOS provides a broad range of interoperable capability for discovering, binding to and interrogating individual sensors, sensor platforms, or networked constellations of sensors in real-time, archived or simulated environments.

Ref: http://www.opengeospatial.org/standards/sos

1.1.8.5 Sensor Planning Service (SPS)

The Sensor Planning Service (SPS) is a service by which a client can determine collection feasibility for a desired set of collection requests for one or more sensors/platforms, or a client may submit collection requests directly to these sensors/platforms.

The targeted users of the SPS are enterprises that need to automate complex information flows, and depend on live and stored data from sensors and imaging devices. In such environments, specific information requirements give rise to frequent and varied collection requests. Quickly getting an observation from a sensor at the right time and place may be critical, and getting data that was collected at a specific place at a specific time in the past may also be critical. SPS provides standard interfaces for determining the feasibility of an intended sensor planning request, for submitting such a request, for inquiring about the status of such a request, for updating or cancelling such a request.

Ref: http://www.opengeospatial.org/standards/sps

1.1.8.6 SWE Common Data Model (SWE Common)

SWE Common Data Model (SWE Common) is a self-describing data model for transferring data in an unambiguous fashion; it is built in accord with ISO 11404 standard, which specifies the nomenclature and shared semantics for a collection of data types commonly occurring in programming languages and software interfaces.

SWE Common supports XML and ASCII as well as Binary encodings; it allows encryption and compression, and supports various native formats, common to all encodings and services. More precisely, SWE Common specification defines standard data types, such as Quantity, Category, Boolean, Count, Time, DataArray, and DataRecord. The DataArray provides means of packaging large volumes of data into an XML envelope using ASCII, raw binary or even compressed binary encodings; it is of special interest for high data throughput systems since it allows tempering the impact of XML bloated nature while preserving interoperability.

Encoding in SWE Common demonstrates a number of advantages for geo-spatial applications over other XML dialects:

- encoding truly represents a conceptual model;
- encoding provides information required for accurate distinguishing of 2D data sets, e.g. that a data set is of a grid type or a trajectory;
- encoding is simple and it is easy to create software to publish and/or read the data as well as export the data to other applications;
- encoding is optimized for minimum size, thus reducing time needed to access and/or process the data.

SWE Common is a result of OGC work to merge O&M and SensorML data models. The two sets of data models were combined and extended to provide a common data model that is now used not only in SensorML and O&M, but throughout the SWE specifications.

Ref: http://www.ogcnetwork.net/SWE_Common

http://www.oostethys.org/ogc-oceans-interoperability-experiment/topics/swecommon

1.2 SOA Related Standards

Service Oriented Architecture (SOA) is an architectural style whose goal is to achieve loose coupling among interacting software agents. A service is a unit of work done by a service provider to achieve desired end results for a service consumer. Both provider and consumer are roles played by software agents on behalf of their owners. Loose coupling refers to the interfaces that are independent of each other's implementation. In a loosely coupled system, it should be possible to swap-out one of the components and replace it with another without effect to the overall system.

In addition, SOA services allow dynamic discovery and invocation and provide platform-agnostic interfaces. Dynamic discovery implies that some sort of registry is in place where these services are listed and which incorporates a lookup function. The platform-agnostic interface means that a client on any platform (OS, language, hardware) can discover and consume the service.

Currently, Web services are almost exclusively considered as the standard way to implement SOA services. It is generally accepted that a Web service is SOA if it meets at least the following requirements:

1. Interfaces must be based on Internet protocols such as HTTP, FTP, and SMTP;
2. Messages must be in XML with the exception of binary data attachments.

There are two main styles of Web services: SOAP services and REST services.

However, there can be a Service-Oriented Architecture without Web services, or even XML. According to the definition, all distributed computing technologies that have a concept of services, are defined by interfaces, and are platform agnostic, e.g. CORBA, DCE, DCOM, RMI, can be considered SOA.

Ref: http://webservices.xml.com/pub/a/ws/2003/09/30/soa.html

http://soaprpc.wordpress.com/category/web-services/

1.2.1 Hyper Text Transfer Protocol (HTTP) and HTTP Secure (HTTPS)

1.2.1.1 HTTP

The Hypertext Transfer Protocol (HTTP) is an application-level protocol for distributed, collaborative, hypermedia information systems. It is a generic, stateless, protocol which can be used for many tasks beyond its use for hypertext, such as name servers and distributed object management systems, through extension of its request methods, error codes and headers. A feature of HTTP is the typing and negotiation of data representation, allowing systems to be built independently of the data being transferred.

HTTP has been in use by the World-Wide Web global information initiative since 1990. The current version of the protocol referred to as "HTTP/1.1" was published as a W3C RFC 2616.

The HTTP is a request/response protocol for a client/server environment. A client making a HTTP request — using a web browser, spider, or other end-user tool — is referred to as the user agent. The responding server—which stores or creates resources such as HTML files and images — is called the origin server. In between the user agent and origin server may be several intermediaries, such as proxies, gateways, and tunnels. The client sends a request to the server in the form of a request method, URI, and protocol version, followed by a message containing request modifiers, client information, and possible body content over a connection with a server. The server responds with a status line, including the message's protocol version and a success or error code, followed by a message containing server information, entity metadata, and possible content.

In HTTP/1.0, most implementations opened a new connection for each request/response exchange. In HTTP/1.1, a connection may be used for more than one request/response exchanges.

Since modern information systems require more functionality than simple retrieval, including search, front-end update, and annotation, the HTTP/1.1 allows an open-ended set of methods and headers that indicate the purpose of a request. It builds on the discipline of reference provided by the Uniform Resource Identifier (URI), in form of a location (URL) or name (URN), for indicating the resource to which a method is to be applied.

Currently, HTTP is widely and commonly used as a generic protocol for communication between user agents and proxies/gateways to other Internet systems, including those supported by the SMTP, NNTP, FTP, Gopher, WAIS and many other protocols. In this way, HTTP allows hypermedia access to resources available from diverse applications.

HTTP communication usually takes place over TCP/IP connections; the default port is TCP 80, but other ports can also be used. However, HTTP can be implemented on top of any other protocol – HTTP's only requirement is a reliable transport; any protocol that provides such guarantees can be used.

Ref: http://www.w3.org/Protocols/

1.2.1.2 HTTPS

HTTP over SSL or HTTP Secure (HTTPS) is a combination of the HTTP in conjunction with a cryptographic protocol of some sort. HTTPS uses either Secure Socket Layer (SSL) or Transport Layer Security (TLS) as a sublayer under regular HTTP application layering. Originally, HTTPS was only used with SSL encryption, but this has become obsolete upon development of TLS. HTTPS encrypts and decrypts user request as well as the server response. The use of HTTPS protects against eavesdropping and man-in-the-middle attacks.

HTTPS was developed by Netscape Communications in 1994 for its Netscape Navigator Web browser. HTTPS was adopted as a web standard with the publication of RFC 2818 by the Internet Engineering Task Force (IETF) in May 2000.

HTTPS and SSL support the use of X.509 digital certificates from the server so that, if necessary, a user can authenticate the sender. Unless a different port is specified, HTTPS uses port 443 instead of HTTP port 80 in its interactions with the lower layer, TCP/IP.

The effectiveness of HTTPS can be limited by poor implementation of browser or server software or a lack of support for some algorithms. Furthermore, although HTTPS secures data as it travels between the server and the client, once the data is decrypted at its destination, it is only as secure as the host computer.

Ref: http://tools.ietf.org/html/rfc2818

1.2.1.3 KVP

The Keyword-Value Pair (KVP) is simply a set of data items that contain a key, e.g. an account number and a value. Keyword-value pairs are widely used in tables and configuration files as well as in the Web Service requests and responses that are sent between client and server using the Hypertext Transfer Protocol (HTTP).

In general, there are two methods of encoding Web Service requests. The first uses XML as the encoding language, the second uses Keyword-value pairs to encode the various parameters of a request. KVP-requests are sent using HTTP GET, while XML-requests have to be sent with HTTP POST. In both cases (XML and KVP), the response to a request or the exception report must be identical. In general KVP requests are shorter, but using KVP for transaction requests is limited.

The examples of a keyword-value pair (KVP) in WFS HTTP GET request, and HTTP POST request encoded in XML are given in the figure below.

```
http://www.someserver.com/servlet/wfs?request=GetFeature&FEATUREID=TEST_BOUNDARY.1000
```

KVP request using HTTP GET

```xml
<?xml version=”1.0”?>
<GetFeature
   Version=”1.1.0”
   service=”WFS”
   xmlns=”http://www.opengis.net/wfs”
   xmlns:ogc=”http://www.opengisnet/ogc”
   xmnls:cad=”http://www.someserver.com/cad”
   xmlns:xsi=”http://www.w3.org/2001/XMLSchema-Instance”
   xsi:schemaLocation=”http://schemas.opengis.net/wfs/1.1.0/wfs.xsd”>
    <Query typeName=”TEST_BOUNDARY”>
         <ogc:Filter>
              <ogc:FeatureId fid=”TEST_BOUNDARY.1000”/>
         </ogc:Filter>
    </Query>
</GetFeature>
```

Same request encoded in XML using HTTP POST

The difference between WFS request encoded in KVP and XML

Ref: Wrapping OGC HTTP-GET/POST Services with SOAP. Discussion Paper. (http://portal.opengeospatial.org/files/?artifact_id=25280)

1.2.2 Java Message Service (JMS)

The Java Message Service (JMS) defines the standard for reliable Enterprise Messaging, which is an essential tool for building enterprise applications. JMS is a set of interfaces and associated semantics that define how a JMS client accesses the facilities of an enterprise messaging product. By combining Java technology with enterprise messaging, the JMS API provides a powerful tool for solving enterprise computing problems by introducing a common API and provider framework that enables the development of portable, message based applications in the Java programming language.
The JMS API improves programmer productivity by defining a common set of messaging concepts and programming strategies that will be supported by all JMS technology-compliant messaging systems.

The JMS API is an integral part of the Java 2, Enterprise Edition (J2EE) platform, and application developers can use messaging with components using J2EE APIs ("J2EE components").

The addition of the JMS API enhances the J2EE platform by simplifying enterprise development, allowing loosely coupled, reliable, asynchronous interactions among J2EE components and legacy systems capable of messaging. A developer can easily add new behavior to a J2EE application with existing business events by adding a new message-driven bean to operate on specific business events.

Ref: http://java.sun.com/products/jms/

1.2.3 eXtensible Markup Language (XML)

The eXtensible Markup Language (XML) is a subset of Standard Generalized Markup Language (SGML), which is standardized by ISO (ISO 8879:1986 SGML). It was designed as an open standard, is licence free, and became a W3C Recommendation in 1998.

XML was designed to transport and store data just as HTML was designed to display data. XML design goal was to represent and exchange data as structured documents across information systems in general, and via the Internet in particular. It is human readable (text based, Unicode support); the XML document conforms to some semantic rules, e.g. Document Type Definition (DTD) or XML Schema (XSD). XML technology is widespread, easily available and inexpensive.

XML can be combined with other data formats, e.g. XML metadata /header and HDF5, NetCDF or BUFR datasets. XML documents can be translated to/from other data formats, and a number of converters have been developed, for example:

- NetCDF to/from XML:
  - (to) NetCDF Markup Language (NcML), and NcML-GML
  - (to&from) LeoNetCDF
- GRIB to XML (via NetCDF):
  - deGRIB + (NetCDF to XML converter)
- HDF5 to XML:
  - d5dump

XML has numerous extensions and specific dialects including, but not limited to, the following:

- Geography Markup Language (GML)
- Digital Weather Markup Language (DWML)
- Climate Data Markup Language (CDML)
- Climate Science Modelling Language (CSML)
- Weather Markup Language (WxML)
- Emergency Data Exchange Language (EDXL)
- Water Markup Language (WaterML)
- Chemical Markup Language (CML),
- Electronic Business XML Initiative (ebXML),
- Scalable Vector Graphics (SVG),
- Sensor Markup Language (SensorML),
- Sensor Web Enablement Common (SWE Common),

Some are further described in this chapter.

The use of XML for data storage and transmission has a lot of incontestable advantages; however, it is a challenge to ensure high performance due to a bloated body of XML messages and overhead of conversion to/from XML.

Several solutions have been proposed to alleviate the performance degradation:

1. since XML is a text format, the compression would be very effective; however, compression takes some processing power, and inevitably affects interoperability although the effect can be mild if the compression protocol is specified in the payload;
2. the use of encoding schemas like SWE Common, which allows wrapping a binary or compressed ASCII data array into an XML envelope;
3. the use of hardware XML accelerators for XML processing would relieve application servers and speed up data processing;
4. the use of SOAP just for negotiation of a specific binary transmission protocol, which is then used for actual data transfer instead of XML;
5. non-traditional approaches like Sun initiative called Fast Web services: Fast Web or just "Fast" uses a binary encoding for the SOAP payload, while the keeping higher level protocols unchanged; it allows using the standard SOAP-XML for development, and switching to the binary protocol for production.

Ref: http://www.w3.org/XML

1.2.3.1 Plain Old XML (POX)

The term Plain Old XML (POX) is used to describe basic XML, as a contrast with complicated, multilayered XML specifications like those for web services or Resource Description Framework (RDF). As a result, in many cases POX is equated with XML implementation with no XML Schema; however, POX is completely compatible with the schema. On the other hand, POX is completely incompatible with SOAP as SOAP is not just using plain XML.

POX is often considered as complementary to REST, since REST generally refers to a communication pattern, while POX refers to an information format style. However, it is not exactly right: REST is an architectural style that is compatible with XML but independent from it. REST is identified by resource representations, a uniform interface, and linking; so there can be RESTful POX applications as well as other RESTful applications that do not use POX, e.g. RESTful SOAP applications.

Ref: http://msdn.microsoft.com/en-us/library/aa395208.aspx

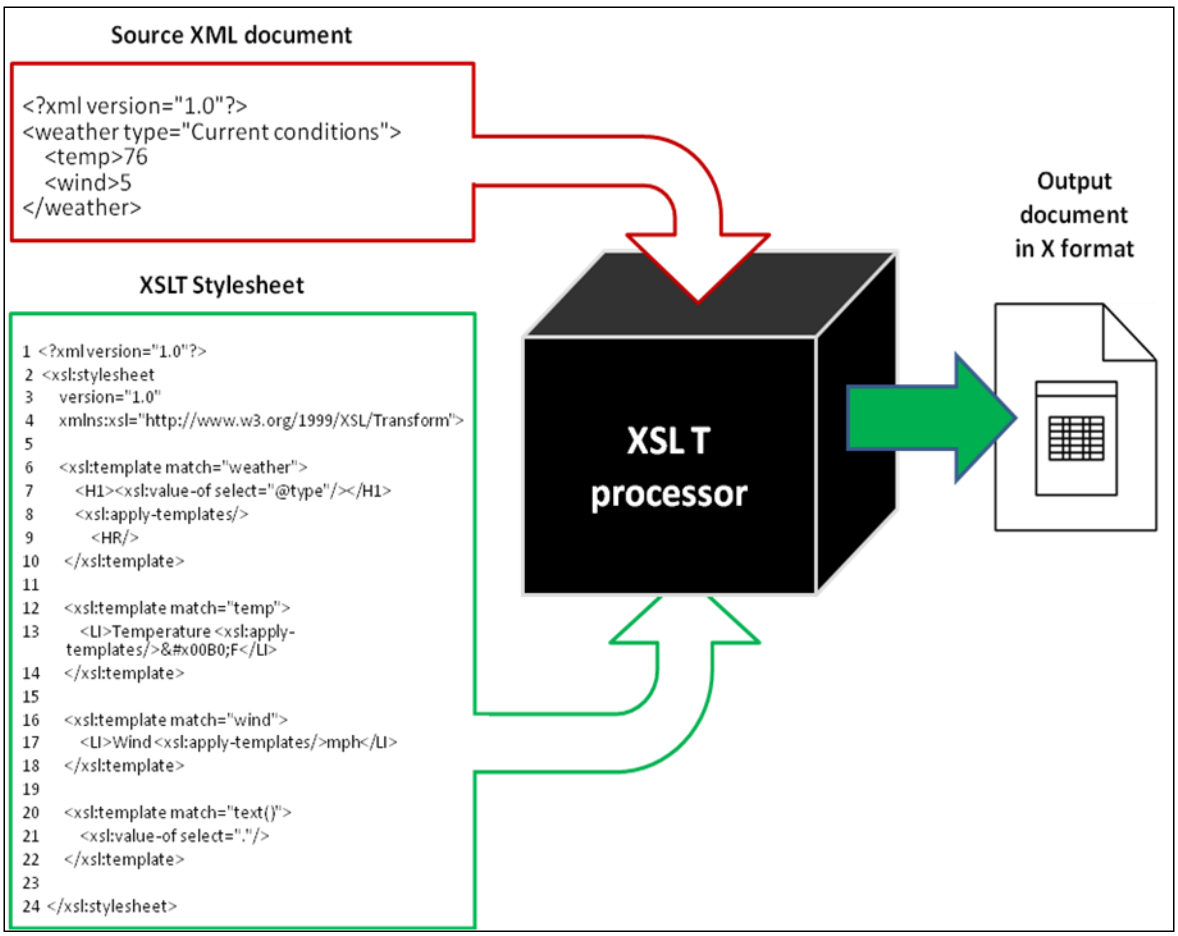
1.2.3.2 eXtensible Stylesheet Language Transformations (XSLT)

Extensible Stylesheet Language Transformations (XSLT) is an XML-based language that provides a way of transforming XML documents from one structure to another. XSLT can be used to create HTML, so XML document can be viewed in a generic Web browser, or XSLT can be used to transform XML document to any other XML structure, or even non-XML structures, e.g. CSV file to open with MS Excel.

XSLT is not a procedural programming language like C or Java, but rather a declarative language based on rules; rules are self-contained objects that do something as soon as they are executed. XSLT rules are called "templates", and are specified using the <xsl:template> element. A template will only execute when an event occurs in the XML document being transformed; the event that the template is to catch is defined using the match attribute on the <xsl:template> start tag.

Each template contains processing instructions to tell the XSLT processor how to treat the object that triggered the rule. An XML document that describes a collection of templates, and guides the XSLT processor in the production of the output document, is known as XSLT stylesheet. The transformation process is illustrated in the following figure:

XSLT Process Diagram



Ref: http://www.w3.org/TR/xslt

1.2.4 Web Services Description Language (WSDL)

The Web Services Description Language (WSDL) provides an XML grammar for describing the Web service details, i.e. the location of the service and the operations that the service exposes. It also provides a way to define bindings for each interface and protocol combination along with the endpoint address for each one. A complete WSDL definition contains all of the information necessary to invoke a Web service.

WSDL and XML schema (XSD) are most commonly used to describe SOAP Web services; it helps to ensure interoperability at the service description layer.

A WSDL document defines services as collections of network endpoints, or ports. In WSDL, the abstract definition of endpoints and messages is separated from their concrete network deployment or data format bindings. This allows the reuse of abstract definitions: messages, which are abstract descriptions of the data being exchanged, and port types which are abstract collections of operations. The concrete protocol and data format specifications for a particular port type constitute a reusable binding. A port is defined by associating a network address with a reusable binding, and a collection of ports define a service. Hence, a WSDL document uses the following elements in the definition of network services:

- Types – a container for data type definitions using some type system (such as XSD).
- Message – an abstract, typed definition of the data being communicated.
- Operation – an abstract description of an action supported by the service.
- Port Type – an abstract set of operations supported by one or more endpoints.
- Binding – a concrete protocol and data format specification for a particular port type.
- Port – a single endpoint defined as a combination of a binding and a network address.
- Service – a collection of related endpoints.

In addition, WSDL defines a common binding mechanism. This is used to attach a specific protocol or data format or structure to an abstract message, operation, or endpoint. It allows the reuse of abstract definitions. WSDL specifically describes binding extensions for SOAP, HTTP GET / POST and MIME. However, nothing precludes the use of other binding extensions with WSDL.

Since WSDL is a machine-readable language (e.g., it's just an XML file), tools and infrastructure can be easily built around it: developers can use WSDL definitions to generate code that knows precisely how to interact with the Web service it describes. This type of code generation hides the tedious details involved in sending and receiving SOAP messages over different protocols and makes implementation of Web services really accessible.

WSDL became a W3C Recommendation in June 2007. The current version WSDL 1.1 is considered the de-facto standard today because of industry-wide support. Most Web services toolkits support WSDL 1.1, but there have been some interoperability problems across the different implementations due to the extensive flexibility of WSDL. The W3C is actively working on the WSDL 1.2, but it's currently only a Working Draft and not supported by the mainstream toolkits, if any.

Ref: http://www.w3.org/TR/wsdl

1.2.5 Simple Object Access Protocol (SOAP)

Simple Object Access Protocol (SOAP) is an XML-based object invocation protocol. SOAP was originally developed for distributed applications to communicate over HTTP and through corporate firewalls.

SOAP defines the use of XML and HTTP to access services, objects and servers in a platform-independent manner.

SOAP does not itself define any application semantics such as a programming model or implementation specific semantics; rather it defines a simple mechanism for expressing application semantics by providing a modular packaging model and encoding mechanisms for encoding data within modules. This allows SOAP to be used in a large variety of systems. SOAP consists of three parts:

- SOAP envelope construct defines an overall framework for expressing what is in a message; who should deal with it, and whether it is optional or mandatory
- SOAP encoding rules defines a serialization mechanism that can be used to exchange instances of application-defined data types
- SOAP RPC representation defines a convention that can be used to represent remote procedure calls and responses.

In addition to the above, SOAP also defines two protocol bindings that describe how a SOAP message can be carried in HTTP messages either with or without the HTTP Extension Framework.

A SOAP Web service is the most common and marketed form of Web service in the industry; to be SOA-compliant it needs to meet the following additional requirements:

1. except for binary data attachment, messages must be carried by SOAP;
2. the service must be described in Web Service Definition Language (WSDL).

Ref: http://www.w3.org/2000/xp/Group/

1.2.6 Representational State Transfer (REST)

A Representational State Transfer (REST) Web service is an SOA based on the concept of "resource". A resource is anything that has a Uniform Resource Identifier (URI). A REST web service requires the following additional constraints:

1. Interfaces are limited to HTTP, and the following semantics are defined:
    a. HTTP GET is used for obtaining a representation of a resource.
    b. HTTP DELETE is used for removing representations of a resource.
    c. HTTP POST is used for updating or creating the representations of a resource.
    d. HTTP PUT is used for creating representations of a resource.
2. Most messages are in XML, confined by a schema written in a schema language such as XML Schema from W3C or RELAX NG.
3. Simple messages can be encoded with URL encoding.
4. Service and service providers must be resources while a consumer can be a resource.

REST web services require little infrastructure support apart from standard HTTP and XML processing technologies, which are now well supported by most programming languages and platforms. REST web services are simple and effective because HTTP is the most widely available interface, and it is good enough for most applications. In many cases, the simplicity of HTTP simply outweighs the complexity of introducing an additional transport layer.

While the SOAP/WSDL/BPEL approach to Web services is still the most widely used ways to implement SOA, the REST is also carving its way. In fact, some Web services may not need the extra layer of description provided by SOAP and WSDL to ensure reliable workflows. For example, OGC is gradually adjusting the Web service specifications to accommodate a REST-based architecture; there is a REST API for the OGC SOS, and the latest version of the OGC WMS draft standard defines a REST interface.

Ref: http://www.xfront.com/files/rest.html

1.2.7 JMBL Web Services

Joint Meteorology and Oceanography (METOC) Broker Language (JMBL) is a specification for a standard language that brokers the exchange of information between METOC data providers and user applications. The way this information is exchanged is with a Web service.

The JMBL Web Service is a SOAP Web Service, where the request and response are Extensible Markup Language (XML) messages embedded in SOAP messages that are exchanged between the client and service. Invoking the JMBL Web service refers to the actions that a client application performs to use the JMBL Web Service. Client applications that invoke Web Services can be written using any technology, e.g. Java, C++, Perl, Microsoft SOAP Toolkit, .NET, etc. The only requirement is that the SOAP front end must wrap the JMBL request and the JMBL request must be compliant to the JMBL XML schemas to gain access to the JMBL Web Service data.

Based on the user's request, the SOAP response contains a JMBL XML response with either embedded data values as part of the XML or a reference to an external file of data encoded in various formats, e.g. GRIB, GIF, JPEG, etc. That allows the SOAP response to return instead of large files just a URL that the data can be retrieved from via any file delivery protocol, e.g. FTP/SFTP, or HTTP/HTTPS.

Ref: https://afweather.afwa.af.mil/prodcap.html

https://wiki.ucar.edu/display/NNEWD/JMBL

1.2.8 OASIS Models and Languages

1.2.8.1 Reference Model for SOA

Organization for the Advancement of Structured Information Standards (OASIS) was founded in 1993 under the name SGML Open as a consortium of vendors and users devoted to developing guidelines for interoperability among products that support the Standard Generalized Markup Language (SGML). OASIS changed its name in 1998 to reflect an expanded scope of technical work, including XML and other related standards. OASIS is a not-for-profit consortium that drives the development, convergence and adoption of open standards for the global information society. The consortium produces more Web services standards than any other organization along with standards for security, e-business, and standardization efforts in the public sector and for application-specific markets. Founded in 1993, OASIS has more than 5,000 participants representing over 600 organizations and individual members in 100 countries.

OASIS Reference Model for Service Oriented Architecture is an abstract framework for understanding significant entities and relationships between them within a service-oriented environment, and for the development of consistent standards or specifications supporting that environment. It is based on unifying concepts of SOA and may be used by architects developing specific service oriented architectures or in training and explaining SOA. A reference model is not directly tied to any standards, technologies or other concrete implementation details. It does seek to provide a common semantics that can be used unambiguously across and between different implementations. The relationship between the Reference Model and particular architectures, technologies and other aspects of SOA is illustrated in Figure 1.1.3. While service-orientation may be a popular concept found in a broad variety of applications, this reference model focuses on the field of software architecture. The concepts and relationships described may apply to other "service" environments; however, this specification makes no attempt to completely account for use outside of the software domain.

OASIS Reference Model defines 7 principal concepts:

- Service
- Visibility
- Interaction
- Real world effect
- Service description
- Contract and Policy
- Execution context

A **service** is a mechanism to enable access to one or more capabilities; it is the cornerstone of the model.

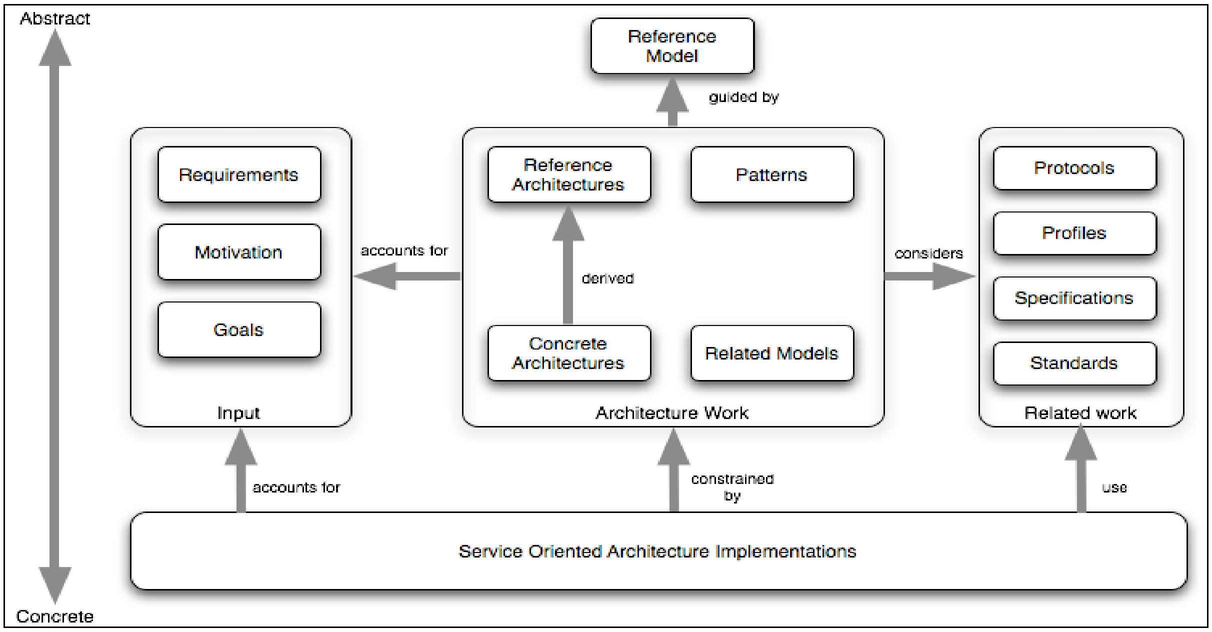
Visibility, Interaction and **Real world effect** describe the dynamics of services.

Service description, Contract and Policy and **Execution** context deal with describing services:

The following diagram illustrates the relationship between the Reference Model and other aspects of SOA:[2]

[2] Taken from OASIS Reference Model for Service Oriented Architecture 1.0

Relational Diagram Between the Reference Model and Other Aspects of SOA



Ref: http://docs.oasis-open.org/soa-rm/v1.0/soa-rm.pdf

1.2.8.2 Web Services Business Process Execution Language (WS-BPEL)

Web Services Business Process Execution Language (WS-BPEL) provides a language for formally describing business processes and business interaction protocols. WS-BPEL is an OASIS standard; WS-BPEL 2.0 was approved in April 2007.

WS-BPEL defines a model and a grammar for describing the behavior of a business process based on interactions between the process and its partners; the interactions occur through Web services interfaces exclusively. The process defines how multiple service interactions with these partners are coordinated to achieve a business goal, as well as the state and the logic necessary for this coordination.

WS-BPEL process definition can be deployed in different ways and in different scenarios, while maintaining a uniform application-level behavior across all of them. WS-BPEL also introduces systematic mechanisms for dealing with business exceptions.

WS-BPEL separates the public aspects of business process behavior from internal or private aspects--and supports both. The standard can be used both for executable processes, which describe the actual behavior of participants in business interactions, and for abstract processes, that may be used to represent publicly observable behaviors. Abstract processes serve a descriptive role and allow for more than one possible use case.

By providing a language for specifying both executable and abstract business processes, BPEL extends the Web services interaction model to help better support business-to-business transactions. This protects business partners from the need to reveal all their internal decision making and data management to one another. Separating public from private processes also provides companies with the freedom to change confidential aspects of the process implementation without affecting the observable behavior.

WS-BPEL leverages other Web services standards such as SOAP and WSDL for communication and interface description. By describing the inbound and outbound process interfaces in WSDL, BPEL enables them to be easily integrated into other processes or applications. In turn, this allows consumers of a process to inspect and invoke a BPEL process just like any other Web service, thereby inheriting all other aspects of a Web service such as quality of service policies. Any web service with a WSDL 1.1 contract can be used by or within by a BPEL process.

Although WS-BPEL 2.0 is not designed to use REST services (because these types of services do not use WSDL) there are ways of doing that. One way to work with a REST service in BPEL would be to generate WSDL/Schema that describes the REST interactions and then create an adapter service or other means to communicate with the REST endpoint.

WS-BPEL 2.0 was not designed for human workflow, and does not deal with human interactions directly; however, a proposed extension to WS-BPEL 2.0 addresses this issue. The BPEL4People extension aims to model human interactions as first class citizens within a BPEL process through the use of an extension activity.

Ref: http://www.oasis-open.org/specs/#wsbpelv2.0

1.2.8.3 Universal Discover, Description, and Integration (UDDI)

In order for a service to be used or reused a user has to be able to discover it. A registry is a SOA way of publishing services and helping potential users to find the services they might be interested in. The simplest implementation of the registry is nothing more than a library index. The registry should offer both search and browse interfaces, and should be organized logically to facilitate quick and accurate discovery of services.

The Universal Discover, Description, and Integration (UDDI) is an approved OASIS Standard and a key member of the Web services stack; it provides a data model and a set of interfaces (all Web services themselves) for publishing and discovering services, as well as a further set of interfaces for managing the registry server itself.

The UDDI specification relies on several other established industry standards, including HTTP, XML, XML Schema (XSD), SOAP, and WSDL. The UDDI specifications define:

- SOAP APIs that applications use to query and to publish information to a UDDI registry;
- XML Schema schemata of the registry data model and the SOAP message formats;
- WSDL definitions of the SOAP APIs;
- UDDI registry definitions of various identifier and category systems that may be used to identify and categorize UDDI registrations.

Although UDDI used to be a cornerstone of the registry, the original concept of registry has evolved quite a bit further to a more complete repository, adopting modern technologies like policy-based filtering and sophisticated run-time governing solutions. Beyond registry is the whole concept of policy and metadata services that provide comprehensive repositories for all design-time and run-time artifacts for the services that make up the SOA.

Ref: http://uddi.xml.org/

1.2.8.4 Electronic Business XML (ebXML) Registry

The Electronic Business (eBusiness) XML (ebXML) set of specifications enable electronic trading relationships between business partners. The ebXML supports the substantial elements of a typical B2B infrastructure:

- XML-based messaging
- Trading partner agreements
- Registry

The ebXML is developing by the United Nations Centre for Trade Facilitation and Electronic Business (UN/CEFACT) and OASIS, with additional support from a multitude of industry leaders.

The ebXML Registry Services specification defines the interface to the ebXML Registry Services as well as interaction protocols, message definitions and XML schema.

An ebXML Registry Information Model (ebRIM) specification describes entities that store information in an ebXML registry, types of metadata that are stored in the SOA Registry, and SOA Repository implementation options.

An SOA registry is a resource that enterprises share to publish, discover, and consume Web services. Content such as XML Schemas, Document Type Definitions (DTDs), and Web Services Description Language (WSDL) documents, can be kept in an SOA Repository, which is then used by SOA Registry to enable a subscribe/publish model for the services. In that way, an SOA Repository is a mere storage that keeps information published to an SOA Registry.

Per the ebRIM specification, the SOA Repository may be implemented within an ebXML Registry in the form of a relational database schema, object database schema, or some other physical schema.

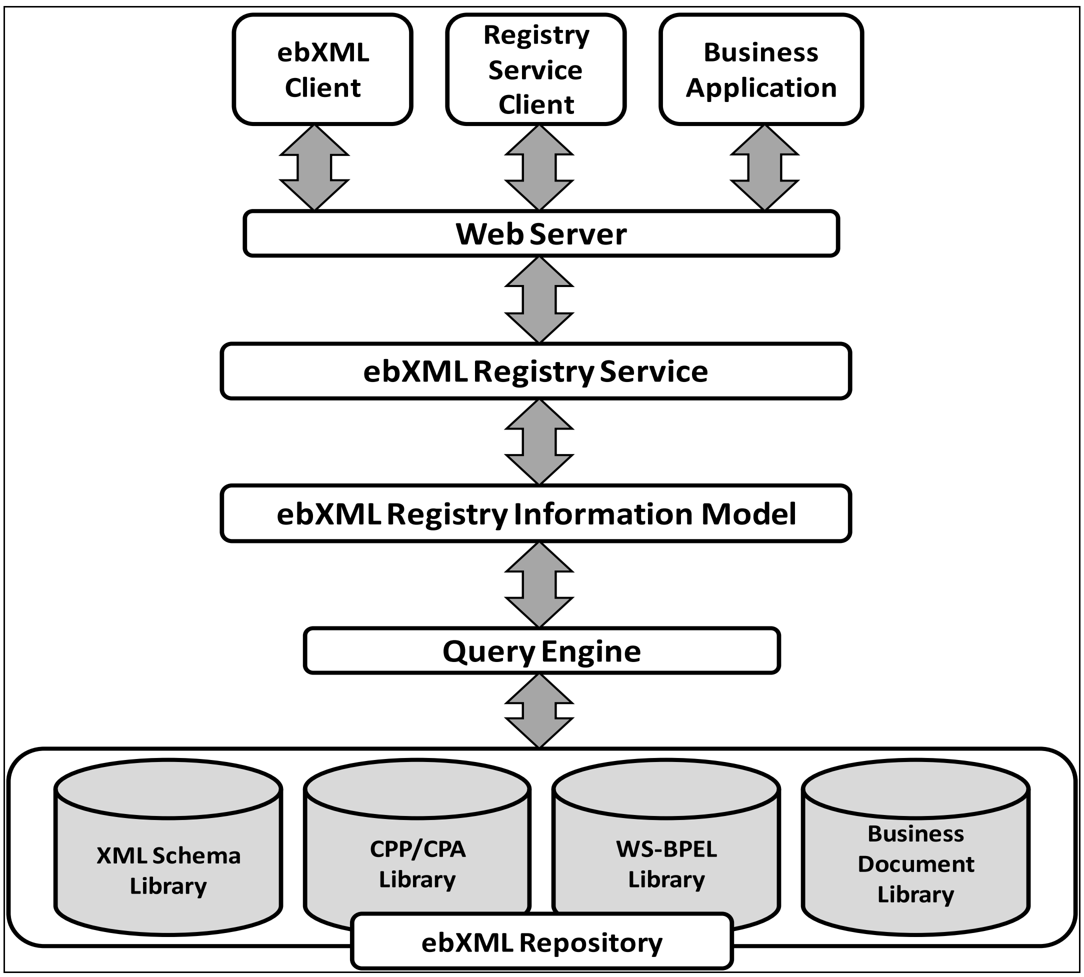
The ebXML Registry is a successor of UDDI; both UDDI and ebXML standards enable enterprises to discover each other, exchange XML-based messages, and engage in meaningful collaborations based on their individual trading-partner agreements and public business processes. Like UDDI, ebXML Registry facilitates seamless and automatic integration between trading partners and supports the overall goal and vision of an SOA in which applications communicate and share functionality without human interaction.

By contrast, ebXML Registry standard introduces a concept of integrated repository to store content as well as metadata; the protocols and information model is generic and extensible; it supports multi-registry topologies using loosely coupled federations. As compared with UDDI, ebXML registry provides a number of benefits:

- discovery and maintenance of registered content;
- support for collaborative development, where users can create XML content and submit it to the registry for use and potential enhancement by the authorized parties;
- persistence of WS-BPEL, WSDL, and business documents during interactions between trading partners;
- secure version control of registered content;
- federation of cooperating registries to provide a single view of registered content by seamless querying, synchronization, and relocation of registered content;
- event notification via email or Web services.

The high-level ebXML Registry flowchart is shown in the figure below

Example ebXML Registry Outline Flowchart



Ref: http://www.ebxml.org/specs/index.htm

http://www.ebxml.org/specs/ebRS.pdf

http://www.sun.com/products/soa/registry/soa_registry_wp.pdf

1.3 Metadata standards

Metadata is required to allow discovering, evaluating, and accessing of data resources. Metadata is applicable to services, datasets, geospatial features, and sensors.

Metadata comes in a variety of forms. Some of the more common categories of metadata are:

- Structural Metadata, which describes the dataset composition (e.g., data encoding, file type) and relationships between specific parts of the dataset.
- Data Access Metadata, which is a descriptive part of metadata associated with the mechanism needed to acquire a dataset and describing it. In case that mechanism is a service than it qualifies as Service Metadata; this metadata includes information required for determination if a service is of interest for user and how the service may be invoked. The service metadata becomes available through GetCapabilities function.
- Security Metadata that defines the information (e.g., security and privacy markings consistent with applicable standards) through which systems will be able to control access to data.
- Sensor Metadata, which describes individual sensors, platforms and sensor networks. It provides user with the information of sensor's capabilities, e.g. location, range, accuracy, and helps to determine a value of the data collected by the sensors.
- Discovery Metadata – a portion of the metadata that is used to locate the required data sets.
- QA/QC Metadata, which represents any information pertaining to a data quality and reliability, or any other data quality assessment details; some level of quality of data information should be made publicly available so that data users are aware of the level of data validity.
- Other Metadata that may include vocabularies, taxonomic structures used for organizing data assets, interface specifications, mapping tables, author and source description.

Metadata standards have evolved considerably over the last several decades driven by the needs of the communities that use these standards.

The Directory Interchange Format (DIF) emerged in 1987 as an answer to the need of data discovery, and has continued to evolve since that time.

In early 1990's the Federal Geographic Data Committee (FGDC) created the Content Standard for Digital Geographic Metadata (CSDGM), also known as the FGDC Metadata Standard. The FGDC Clearinghouse System was created to provide a distributed search of FGDC metadata.

NOAA created a set of NOAA specific extensions, i.e. "NOAA Supplemental" as an addition to the FGDC Metadata Standard. At the same time, the NASA Earth Observing System (EOS) was developing metadata content guidelines for the remotely sensed data to be collected as part of that project. Both were eventually superseded by the development and approval of the FGDC Remote Sensing Extensions (RSE).

Metadata standards development expanded into a more international effort during the late 1990’s with the development of an extensive set of standards for geographic data by the International Organization for Standardization (ISO) Technical Committee 211.

1.3.1 Directory Interchange Format (DIF)

The Directory Interchange Format (DIF) is a relatively minimal metadata standard; it is a "container" for the metadata elements that are maintained in the CEOS International Directory Network (IDN) database, where validation for mandatory fields, keywords, and personnel takes place.

The DIF is used to create directory entries which describe a group of data. The DIF consists of a collection of fields which detail specific information about the data; it has a total of 36 fields, including 8 mandatory fields. The mandatory elements are EntryID, Entry Title, Keywords, ISO Topic Category, Data Center, Summary, Metadata Name and Metadata Version. Some of the fields are text fields; others require the use of controlled keywords (sometimes known as "valids"). The DIF standard is compatible with both the ISO 19115 and CSDGM standards.

The DIF allows users of data to understand the contents of a data set and contains those fields which are necessary for users to decide whether a particular data set would be useful for their needs.

Ref: http://gcmd.nasa.gov/User/difguide/difman.html

1.3.2 NASA's Global Change Master Directory (GCMD)

The Global Change Master Directory (GCMD) enables users to locate and obtain access to Earth science data sets and services relevant to Earth science research. The GCMD database holds more than 25,000 descriptions of Earth science data sets and services covering all aspects of Earth and environmental sciences.

The GCMD maintains standard vocabularies in eleven categories. All eleven are proposed for use in NOAA in order to maintain consistent vocabularies across NOAA. Many significant applications within NOAA are using the standard, e.g. NOAA Metadata Manager and Repository (NMMR), NOAA Comprehensive Observational Requirements List (CORL) and many NOAA metadata management systems.

Ref: http://gcmd.gsfc.nasa.gov/index.html

1.3.3 Federal Geographic Data Committee (FGDC)

The Federal Geographic Data Committee (FGDC) is an interagency committee that promotes the coordinated development, use, sharing, and dissemination of geospatial data on a national basis. The FGDC is tasked by Executive Order 12906 to develop procedures and assist in the implementation of a distributed discovery mechanism for national digital geospatial data. Geospatial metadata are critical to data discovery, and are used to document geographic digital resources such as Geographic Information System (GIS) files, geospatial databases, and earth imagery. A metadata record is a file of information, usually presented as an XML document, which captures the basic characteristics of a data or information resource. It represents the “who”, “what”, “when”, “where”, “why” and “how” of the resource. A geospatial metadata record includes core library catalog elements such as Title, Abstract, and

Publication Data; geographic elements such as Geographic Extent and Projection Information; and database elements such as Attribute Label Definitions and Attribute Domain Values.

Ref: http://www.fgdc.gov/

1.3.3.1 Content Standard for Digital Geospatial Metadata (CSDGM)

The FGDC Content Standard for Digital Geospatial Metadata (CSDGM) was developed from the perspective of defining the information required by a prospective user. This includes such information as determining the availability of a set of geospatial data, determining the fitness of the set of geospatial data for an intended use, determining the means of accessing the set of geospatial data, and successfully transferring the set of geospatial data. As such, the standard provides a common set of terminology and definitions for the documentation of digital geospatial data. The standard establishes the names of data elements and compound elements (groups of data elements) to be used for documentation, the definitions of these compound elements and data elements, and the information about the values that are to be provided for the data elements. The standard does not specify the means by which this information is organized in a computer system or in a data transfer, or the means by which this information is transmitted, communicated, or presented to the user.

FGDC originally adopted the Content Standard for Digital Geospatial Metadata (CSDGM) in 1994 and revised it in 1998. According to Executive Order 12096 all Federal agencies are ordered to use this standard to document geospatial data created as of January, 1995. The standard is often referred to as the 'FGDC Metadata Standard' and has been implemented beyond the federal level with State and local governments adopting the metadata standard as well.

A key feature of the CSDGM Version 2 is the ability of geospatial data communities to customize the base CSDGM. Extensions are a set of added elements that extend the standard to better serve the community or data type. Profiles are custom adaptations of the standard that may specify specific domain values for existing CSDGM elements and/or increase conditionality of a specific element. Profiles may also included extensions.

FGDC Endorsed ***Extensions*** to the CSDGM Version 2 (FGDC-STD-001-1998):

- **Content Standard for Digital Geospatial Metadata: Extensions for Remote Sensing Metadata** - Extended elements to support the documentation of geospatial data directly obtained from remote sensing. This extension includes elements that describe the remote sensing platform and sensors. This extension is intended for the documentation of data collected directly from the sensor. It is not intended for the documentation of data derived from remotely sensed data such as classified imagery. The core CSDGM standard should be used to document derived data.

FGDC Endorsed ***Profiles*** of the CSDGM Version 2 (FGDC-STD-001-1998):

- **Biological Data Profile of the Content Standard for Digital Geospatial Metadata** - The profile broadens the application of the CSDGM so that it is more easily applied to data that are not explicitly geographic (laboratory results, field notes, specimen collections, research reports) but can be associated with a geographic location. The profile changes the conditionality and domains of CSDGM elements; requires the use of a specified taxonomical vocabulary, and adds elements.

- **Metadata Profile for Shoreline Data** - The profile addresses variability in the definition and mapping of shorelines by providing a standardized set of terms and data elements required to support metadata for shoreline and coastal data sets. The profile also includes a glossary and bibliography.

The international community, through the International Organization of Standards (ISO), has developed and approved an international metadata standard, ISO 19115. This international standard provides a common framework for producing and exchanging geographic metadata between nations. As a member of ISO, the US required to revise the CSDGM in accord with ISO 19115.

Each nation can craft their own profile of ISO 19115 with the requirement that it include the 13 core element. In this context, the United States of America and Canada have agreed to revise their respective metadata standards and develop a common profile of ISO19115. North American Profile of ISO19115 (NAP – Metadata) will enhance interoperability of geographic information metadata in North America.

Ref: http://www.fgdc.gov/metadata/geospatial-metadata-standards

1.3.3.2 NSSDA

The National Standard for Spatial Data Accuracy (NSSDA) implements a well-defined statistic and testing methodology for positional accuracy of maps and geospatial data derived from sources such as aerial photographs, satellite imagery, or maps. Accuracy is reported in ground units. The testing methodology is comparison of data set coordinate values with coordinate values from a higher accuracy source for points that represent features readily visible or recoverable from the ground. While this standard evaluates positional accuracy at points, it applies to geospatial data sets that contain point, vector, or raster spatial objects. Data content standards, such as FGDC Standards for Digital Orthoimagery and Digital Elevation Data, will adapt the NSSDA for particular spatial object representations.

The standard insures flexibility and inclusiveness by omitting accuracy metrics, or threshold values, that data must achieve. However, agencies are encouraged to establish "pass-fail" criteria for their product standards and applications and for contracting purposes. Ultimately, users must identify acceptable accuracies for their applications.

Ref: http://www.fgdc.gov/standards/projects/FGDC-standards-projects/accuracy/part3/index_html/

1.3.4 Geographic Names Information System (GNIS)

The Geographic Names Information System (GNIS) is the Federal Standard for geographic nomenclature. The U.S. Geological Survey developed the GNIS for the U.S. Board on Geographic Names as the official repository of domestic geographic names data; the official vehicle for geographic names use by all departments of the Federal Government; and the source for applying geographic names to Federal electronic and printed products.

The GNIS contains information about physical and cultural geographic features of all types in the United States, associated areas, and Antarctica, current and historical, but not including roads and highways. The database holds the Federally recognized name of each feature and defines the feature location by state, county, USGS topographic map, and geographic coordinates. Other attributes include names or spellings other than the official name, feature designations, feature classification, historical and descriptive information, and for some categories the geometric boundaries.

The database assigns a unique, permanent feature identifier, the Feature ID, as the only standard Federal key for accessing, integrating, or reconciling feature data from multiple data sets. The GNIS collects data from a broad program of partnerships with Federal, State, and local government agencies and other authorized contributors, and provides data to all levels of government, to the public, and to numerous applications through a web query site, web map and feature services, file download services, and customized files upon request.

Ref: http://geonames.usgs.gov/

1.3.5 European Petroleum Survey Group (EPSG) Spatial Reference

A coordinate system (also called a Spatial Reference system) is a means of assigning coordinates to a location and establishing relationships between sets of such coordinates. It enables the interpretation of a set of coordinates as a representation of a position in a real world space.

The European Petroleum Survey Group (EPSG) created a spatial reference database, assigning EPSG unique code numbers (IDs) to spatial reference systems (combinations of coordinate systems, projections, and datums).

The ESPG has transformed into the Oil & Gas producers (OGP) Surveying and Positioning Committee (OGP SPC), which via the Geodesy Subcommittee keeps maintaining and publishing a dataset of parameters for coordinate reference system and coordinate transformation description.

The EPSG Geodetic Parameter Dataset is a repository of parameters required to

- Identify coordinates such that those coordinates describe position unambiguously. This is through a **coordinate reference system (CRS)** definition.
- Define transformations and conversions that allow coordinates to be changed from one CRS to another CRS. Transformations and conversions are collectively called **coordinate operations**.

The EPSG Geodetic Parameter Dataset has been included as reference data in United Kingdom Offshore Operators Association (UKOOA) and Society of Exploration Geophysicists (SEG) positioning data exchange formats, the GeoTIFF interchange format for geo-referenced raster imagery, the IHS Energy Iris21, Public Petroleum Data Model Association (PPDM) and Petrotechnical Open Standards Consortium (POSC) Epicenter data models.

Version 6.14 is the current release of the EPSG dataset, available online through the OGP Geodetic Registry and also distributed in an MS Access 97 database and/or as SQL scripts. It incorporates data received and verified since the release of Version 6.13 in July 2007.

A convenient online version of the database, searchable by keywords or code number, is available now. It also provides additional information for the spatial reference system in a number of other formats:

- Human-Readable OGC Well Known Text (WKT)
- Proj4
- JavaScript Object Notation (JSON)
- GML
- ESRI WKT
- USGS
- MapServer Mapfile
- PostGIS spatial_ref_sys INSERT statement

Ref: http://www.epsg.org/

http://spatialreference.org/

1.3.6 ISO Group of Standards

1.3.6.1 ISO 8601

The ISO 8601 standard defines the ways to represent time and date data; that representation is widely used in the ISO 19xxx standard series. The standard offers representations for the following:

- Date
- Time of the day
- Coordinated universal time (UTC)
- Local time with offset to UTC
- Date and time
- Time intervals
- Recurring time intervals

Representations can be in one of two formats: a basic format that has a minimal number of characters and an extended format that adds characters to enhance human readability. For example, the third of January 2003 can be represented as either 20030103 or 2003-01-03.

ISO 8601 advises numeric representation of dates and times on an internationally agreed basis. It represents elements from the largest to the smallest element: year-month-day:

Calendar date is the most common date representation. It is:

YYYY-MM-DD,

where **YYYY** is the year in the Gregorian calendar, **MM** is the month of the year between 01 (January) and 12 (December), and **DD** is the day of the month between 01 and 31.

Example: 2003-04-01 represents the first day of April in 2003.

Week date is an alternative date representation used in many commercial and industrial applications. It is:

YYYY-Www-D,

where **YYYY** is the Year in the Gregorian calendar, **ww** is the week of the year between 01 (the first week) and 52 or 53 (the last week), and **D** is the day in the week between 1 (Monday) and 7 (Sunday).

Example: 2003-W14-2 represents the second day of the fourteenth week of 2003.

Time of the day is the time representation, using the 24-hour timekeeping system. It is:

hh:mm:ss,

where **hh** is the number of complete hours that have passed since midnight, **mm** is the number of complete minutes since the start of the hour, and **ss** is the number of complete seconds since the start of the minute.

Example: 23:59:59 represents the time one second before midnight.

Date and time represents a specified time of a specified day. When use is made of the calendar date the representation is:

YYYY-MM-DDThh:mm:ss,

where the capital letter T is used to separate the date and time components. Thus, 2003-04-01T13:01:02 represents one minute and two seconds after one o'clock in the afternoon of 2003-04-01.

The standard has provisions for:

a) the omission of components representing smaller units (seconds, minutes), where such precision is not needed
b) the addition of a decimal fraction to the smallest time unit where higher precision is needed.

Ref: http://www.iso.org/iso/catalogue_detail?csnumber=40874

1.3.6.2 ISO 19115

ISO 19115 defines the schema required for describing geographic information and services. It provides information about the identification, the extent, the quality, the spatial and temporal schema, spatial reference, and distribution of digital geographic data. ISO 19115 reflects FGDC, TC 287, ANZLIC and other inputs.

ISO 19115 is applicable to:

- catalogues of datasets, clearinghouse activities, and the full description of datasets;
- geographic datasets, dataset series, and individual geographic features and feature properties.

ISO 19115 defines:

- mandatory and conditional metadata sections, metadata entities, and metadata elements;
- the minimum set of metadata required to serve the full range of metadata applications (data discovery, determining data fitness for use, data access, data transfer, and use of digital data);
- optional metadata elements - to allow for a more extensive standard description of geographic data, if required;
- a method for extending metadata to fit specialized needs.

Although ISO 19115 is applicable to digital data, its principles can be extended on many other forms of geographic data such as maps, charts, and textual documents as well as non-geographic data.

ISO 19115 stipulates for Profiles that can be created in recognition of the fact that the generic standard is too broad; profiles includes a set of restrictions on how the standard is used, and a set of extensions specific to a particular domain.

1.3.6.3 ISO 19115-2 Part 2: Extensions for Imagery and Gridded Data

This complementary standard to ISO 19115 defines metadata elements to support imagery, and gridded data and extends the Unified Modeling Language (UML) model for metadata to include the following:

- it will support the collection and processing of natural and synthetic imagery produced by remote sensing and other imaging processes;
- it will support the collection and processing of geospatial metadata for imagery, gridded and coverage data;
- it will define a data model for information describing geographic imagery and gridded data, establishing the names, definitions, and permissible values for new data elements including new classes relevant to imagery and gridded data.

1.3.6.4 ISO/TS 19139 Metadata – XML Schema Implementation

ISO/TS 19139 defines Geographic Metadata XML (GMD) encoding – an XML Schema implementation derived from ISO 19115. ISO/TS 19139 is designed to provide a common XML specification for describing, validating and exchanging geographic metadata. It is intended to promote interoperability, and exploit ISO 19115's advantages in a concrete implementation specification.

ISO 19139 was issued as a Technical Specification (TS), rather than an "International Standard" as 19115 and others. The document defines a set of XML schemas that are maintained by the TC211 committee; the schemas were developed in accord with the Section 8 of the specification.

Although this specification is directly intended to describe geographic metadata for datasets, the nature of the XML schema allows the schemas defined here to be applied to data sets, aggregations of datasets, geographic features, feature attributes, feature types, and feature attribute types. While the specifics of non-dataset usage of the XML schemas defined here are outside the scope of this specification, these XML schemas are designed to support these types of implementations.

Ref: http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_tc_browse.htm?commid=54904

ftp://www.wmo.int/In-box/To-www/DM/ISO/ISO_19100_intro.doc

1.3.7 WMO Metadata Core Profile

In 2008 WMO defined and endorsed a draft metadata standard WMO Metadata Core Profile v1.1 as a strict subset of ISO 19115:2003. This standard provides a general definition for directory searches and exchange that should be applicable to a wide variety of WMO datasets at the discovery level. Like the fundamental standard ISO 19115, the WMO Core Metadata Standard does neither specify how these metadata should be archived or presented to users nor does specify any particular implementation (this is currently underway with the development of ISO 19139).

One of the challenges facing WMO is the need to represent data in several languages. The Core Profile addresses this by using pre-defined keyword lists, i.e. vocabularies, wherever it can. The vocabularies are available in the WMO Official languages, and it is therefore possible to use one language to search for data that are described in another. Where it is not possible to use vocabularies (such as in the abstract), it is possible to hold versions of the text in different languages. Recognising that it will not be feasible to translate all metadata into all the official languages, the Core Profile requires that any metadata exchanged internationally should contain an English version of these fields.

Ref: http://www.wmo.int/pages/prog/www/WDM/Metadata/documents.html

1.3.8 Nextgen Network-Enabled Weather (NNEW) Metadata Guidelines

The Nextgen Network-Enabled Weather (NNEW) program has developed the set of metadata guidelines based upon ISO 19115, ISO 19119, and ISO 19139. The set covers dataset, organization, and service-related metadata; the guidelines focus on discovery metadata for weather datasets and weather data access services, as well as structural metadata related to weather dataset access. Guidelines for sensor description metadata are also provided, but in less detail.

The guidelines have been developed based on the results of the research and prototyping efforts conducted by the NNEW Program since early 2007.

Ref: https://wiki.ucar.edu/download/attachments/23364539/ATC-354.pdf

1.3.9 Metadata for Climate Models (METAFOR)

METAFOR is an emerging standard being developed by a consortium in Europe but with strong links to the USA. The main objective of METAFOR is to develop a Common Information Model (CIM) to describe climate data and the models that produce it in a standard way, and to ensure the wide adoption of the CIM. The project seeks to address the fragmentation and gaps in availability of metadata as well as duplication of information collection, and problems of identifying, accessing or using climate data that are currently found in existing repositories.

The METAFOR CIM opens the way to:

- enhance data discovery across worldwide climate data repositories;

- develop a greater understanding of the complex processes involved in generating climate data;
- enable the building of sophisticated software and tools to use and interpret the climate data;
- automate the process of climate data retrieval, storage, sharing and preservation as data volumes increase;
- evolve metadata standards to satisfy future requirements in climate research.

METAFOR provides a fairly comprehensive coverage of climate modeling concepts from description of models and model components, the configuration of modeling experiments, derivation of output variables and the physical data outputs. It also includes information on quality (measures and assessments). It has adopted ISO 19115 concepts and in some places the metadata structures directly.

Ref: http://metaforclimate.eu/

1.4 Data Encoding Formats / Data Exchange Standards

1.4.1 Weather Exchange Model (WXXM)

Weather Exchange Model (WXXM) is part of the well-established Aeronautical Information Exchange Model (AIXM). The WXXM is a “meteorological component” of a family of platform (technology) independent, harmonized and interoperable information exchange models designed to cover the information needs of Air Traffic Management (ATM). The WXXM is using GML for compatibility with third-party GML applications, and is following ISO 19100 principles and OGC recommendations.

The WXXM is based on the overarching Weather Information Conceptual Model (WXCM). The WXCM itself is based on the OGC O&M model, and describes the differences and correlations (semantics) of Weather Information for the ATM system.

The first release WXXM was demonstrated in May 2007; it was a proof of concept for the exchange of a limited set of ICAO Annex 3 type of products. The WXXM is under development by EUROCONTROL in coordination with ICAO, the US Federal Aviation Administration (FAA) and the World Meteorological Organization (WMO). The first release of the WXCM was built in the second half of 2007. Release of WXXM version 1.1 was planned for the May 2009 timeframe and is intended to meet the following goals:[3]

- Partition schema (WXXS) into general weather concepts and aviation-specific weather concepts
  - Motivation: Allow users to incorporate only the concepts they need (general schema best practice)
- Align with U.K. Climate Science Modeling Language (CSML) and NetCDF Common Data Model (CDM) general weather data types
- Support the ability to use NetCDF Climate and Forecast (CF) standard names and JMBL parameter names within WXXM.
- Refactor/simplify observation/forecast data type hierarchy

[3] WXXM 1.1 Development and Roadmap presentation by Oliver Newell and Kajal Claypool; AIXM/WXXM Conference; 14 May, 2009

- Refine time semantics
- Motivation: Forecast time semantics in WXXM 1.0 constrained by underlying O&M model
- Evaluate standards-based units of measure (applies to WXXM and AIXM)
- Enhance support for Ontologies

Ref: http://www.eurocontrol.int/aim/public/standard_page/met_wie.html

1.4.2 Joint METOC Broker Language (JMBL)

Joint METOC Broker Language (JMBL) is an XML implementation for the exchange of meteorological and oceanographic (METOC) information. JMBL was developed primarily by the United States Department of Defense to easily communicate between data users and providers in a net-centric environment.

JMBL is an abstraction of the physical database, where the end user is concerned only with data types and weather/oceanographic parameters. The versatility of XML coupled with common naming and formats allows JMBL to be used for transferring data complete with self-defining metadata.

Standardizing the way data is exchanged via JMBL eliminates the need for numerous unique point-to-point interfaces, duplicate storage and translation systems between data stores and the users. The design of JMBL supports the incorporation of new data types, formats, and other choices as needed without breaking compatibility with existing users. Additionally, the use of XML for data and metadata minimizes the need for users to develop specific decoder applications to support the variety of message formats.

The JMBL request structure decomposes meteorological and oceanographic data into ten data types: gridded data, imagery, observations, coded text messages (alphanumeric), climatology, remote sensed observations, product, platform, space weather, and best source. The request structure allows specifying properties that are common to all data types (e.g., time, location, format, distribution method, etc.) as well as data type-specific properties (e.g., forecast period, resolution, image type, etc.).

Similar to the WFS operations **GetCapabilities**, **DescribeFeatureType** and **GetFeature**, the JMBL structure supports three levels of requests:

- a catalogue request returns a listing of the products that are available from an information provider;
- a product request returns products such as WMO bulletins formatted as BUFR, METAR, TAF, etc.
- a specific information element request returns parameters such as temperature, wind speed, and wind direction, among others, along with their associated units of measure and other information necessary to understand the data.

Further evolution of JMBL will allow incorporation of the ISO, OASIS, and OGC standards such as Geographic Markup Language (GML), Common Alerting Protocol (CAP) and Unified Code for Units of Measure.

Ref: https://afweather.afwa.af.mil/prodcap.html https://wiki.ucar.edu/display/NNEWD/JMBL

1.4.3 Common Data Format (CDF)

CDF (Common Data Format) is a conceptual data abstraction for storing, manipulating, and accessing multidimensional data sets. The basic component of CDF is a software programming interface that is a device-independent view of the CDF data model. The application developer is insulated from the actual physical file format for reasons of conceptual simplicity, device independence, and future expandability. CDF files created on any given platform can be transported to any other platform onto which CDF is ported and used with any CDF tools or layered applications.

CDF implementation is a library and toolkit for storing, manipulating, and accessing multi-dimensional data sets. The basic component of CDF is a software programming interface that is a device independent view of the CDF data model.

CDF is ported to and runs on many computer platforms, including such operating systems as Windows, MacOS, Linux, and Solaris.

CDF v3.3 is the current version released on July, 2009. The CDF software, documentation, and user support services are provided by NASA and available to the public free of charge. There are no license agreements or costs involved in obtaining or using CDF.

Ref: http://cdf.gsfc.nasa.gov

1.4.4 Unidata Common Data Model (CDM)

Unidata Common Data Model (CDM) is a project to unify scientific data access. It merges the OPeNDAP, netCDF, and HDF5 data models to create a common API for many types of data. As currently implemented by the NetCDF Java library, it can read (besides OPeNDAP, netCDF, and HDF5) GRIB 1 and 2, BUFR, NEXRAD, and GINI, among others. A pluggable framework allows other developers to add readers for their own specialized formats. The CDM also provides standard APIs for geo-referencing coordinate systems and specialized queries for scientific data types like Grid, Point, and Radial datasets.

Ref: http://www.unidata.ucar.edu/software/netcdf/CDM/

1.4.5 Network Common Data Form (netCDF)

The network Common Data Form (NetCDF) is a set of interfaces for array-oriented (gridded) data access and a freely-distributed collection of data access libraries for C, FORTRAN, C++, Java, and other languages. The netCDF libraries support a machine-independent format for representing scientific data. Together, the interfaces, libraries, and format support the creation, access, and sharing of scientific data.

A netCDF dataset contains dimensions, variables, and attributes, which all have both a name and an ID number by which they are identified. These components can be used together to capture the meaning of data and relations among data fields in an array-oriented dataset. The netCDF library allows simultaneous access to multiple netCDF datasets which are identified by dataset ID numbers, in addition to ordinary file names.

NetCDF is an abstraction that supports a view of data as a collection of self-describing, portable objects that can be accessed through a simple interface. Array values may be accessed directly, without knowing details of how the data are stored. Auxiliary information about the data, such as what units are used, may be stored with the data. Generic utilities and application programs can access netCDF datasets and transform, combine, analyze, or display specified fields of the data. The development of such applications has led to improved accessibility of data and improved re-usability of software for array-oriented data management, analysis, and display.

NetCDF data model demonstrates the following main features:

- self-description: a netCDF file includes information about the data it contains, i.e. metadata;
- portability: a netCDF file can be accessed by computers with different ways of storing integers, characters, and floating-point numbers;
- direct accessibility: a small subset of a large dataset may be accessed efficiently, without first reading through all the preceding data;
- append ability: data may be appended to a properly structured netCDF file without copying the dataset or redefining its structure;
- share ability: one writer and multiple readers may simultaneously access the same netCDF file;
- backward compatibility: access to all earlier forms of netCDF data will be supported by current and future versions of the software.

The netCDF model was based on that of the CDF conceptual model but provided a number of additional features (such as C language bindings, portable to a number of platforms, and machine-independent data format). There is no compatibility between data in CDF and netCDF formats, and as yet no translation software exists to convert data from one format to another.

Some netCDF implementations (e.g., NCSA software) provide an interface to HDF container. With this software, it is possible to use the netCDF calling interface to place data into an HDF file. The netCDF v4 is implementing an enhanced netCDF interface on the HDF storage layer to preserve the desirable common characteristics of netCDF and HDF while taking advantage of their separate strengths: the widespread use and simplicity of netCDF and the generality and performance of HDF.

Ref: http://www.unidata.ucar.edu/software/netcdf/index.html

1.4.5.1 NetCDF Conventions

Since netCDF interface enables but does not require the creation of self-describing datasets, the mere use of netCDF does not guarantee that data is equally meaningful to both humans and machines. If the data comes from a variety of sources, it is essential to adopt some standard, i.e. “convention”, for the metadata so that data can be analyzed and compared more easily. These standards

The purpose of the conventions is to require conforming datasets to contain sufficient metadata that they are self-describing in the sense that each variable in the file has an associated description of what it represents, including physical units if appropriate, and that each value can be located in space (relative to earth-based coordinates) and time. This is the kind of metadata that is used at the time the data is processed and displayed; it can be distinguished from "discovery metadata", which is used in catalogues for identifying datasets. Convention generally provides only rather basic discovery metadata, such as ways to record where and how the file was produced.

An important benefit of a convention is that it enables software tools to display data and perform operations on specified subsets of the data with minimal user intervention. It is possible to provide the metadata describing how a field is located in time and space in many different ways that a human would immediately recognize as equivalent. The purpose in restricting how the metadata is represented is to make it practical to write software that allows a machine to parse that metadata and to automatically associate each data value with its location in time and space. It is equally important that the metadata be easy for human users to write and to understand.

A number of groups have defined their own additional conventions and styles for netCDF data:

- CF Conventions
- COARDS Conventions
- GDT Conventions
- CDC Conventions (for gridded data, compatible with but more restrictive than COARDS)
- NCAR-RAF Conventions for Aircraft Data
- AMBER Trajectory Conventions for molecular dynamics simulations
- ARGO netCDF conventions for data centers
- National Oceanographic Data Center netCDF Conventions
- PMEL-EPIC Conventions
- Unidata Observation Dataset Conventions

1.4.5.1.1 Climate and Forecast (CF) Convention

The CF Convention is intended for use with climate and forecast data, for atmosphere, surface and ocean. The CF Convention standard supports interoperability for earth science data from different sources, and is intended for both model output and observational datasets. The Convention is designed to be backward compatible with the COARDS conventions, which means that a conforming COARDS dataset also conforms to the CF standard; thus new applications that implement the CF Convention will be able to process COARDS datasets.

The CF Conventions follow five general principles.

1. Data should be self-describing, without external tables needed to interpret the file.
2. Conventions should only be developed for known issues, and features should only be added as needed.
3. Conventions should be easy to use, both for data writers and users of data.
4. The metadata and its meaning, as encoded in the conventions, should be readable by humans as well as easily used by programs.
5. Redundancy should be minimized to reduce the chance of inconsistency.

The CF Convention is becoming a widely used standard for atmospheric, ocean, and climate data. It has been adopted by many earth-science projects, e.g. the following:

- AMIP: Atmospheric Model Intercomparison Project
- APE: Aqua-Planet Experiment
- CCMVal: Chemistry-Climate Model Validation Activity
- CEOP: Coordinated Energy and Water Cycle Observation Project
- CFMIP: Cloud Feedback Model Intercomparison Project
- C-LAMP: CCSM Carbon LAnd Model intercomparison Project
- CMIP3: Coupled Model Intercomparison Project (Phase 3)
- C3-Grid: Collaborative Climate Community Data and Processing Grid
- DAMOCLES: Developing Arctic Modeling and Observing Capabilities for Long-term Environmental Studies
- ENSEMBLES
- Godiva2 Data Visualization System
- Gulf of Maine Model Interoperability Project
- Humboldt
- IOOS: Integrated Ocean Observing System
- MERSEA: Marine Environment and Security for the European Area
- MetaMod
- NARCCAP: North American Regional Climate Change Assessment Program
- NERC DataGrid
- NERC RAPID THCMIP (Thermohaline Circulation Model Intercomparison Project)
- PMIP: Paleoclimate Modeling Intercomparison Project
- SAMOS: Shipboard Automated Meteorological and Oceanographic System
- SeaDataNet
- TF HTAP Coordinated Model Studies (TF HTAP = Task Force on Hemispheric Transport of Air Pollutants)

The use of CF Convention is encouraged by many important groups and institutions:

- BADC: British Atmospheric Data Centre
- Climate Limited-area Modelling Community
- GHRSST: Group for High Resolution Sea Surface Temperature
- Gulf of Maine Ocean Data Partnership Modeling Committee
- IFREMER: French Research Institute for Exploitation of the Sea
- OceanSITES
- PCMDI: Program for Climate Model Diagnosis and Intercomparison
- Reading e-Science Centre, UK
- Unidata
- WDCC: World Data Center for Climate

Ref: http://cf-pcmdi.llnl.gov/

1.4.5.2 NetCDF Markup Language (NcML)

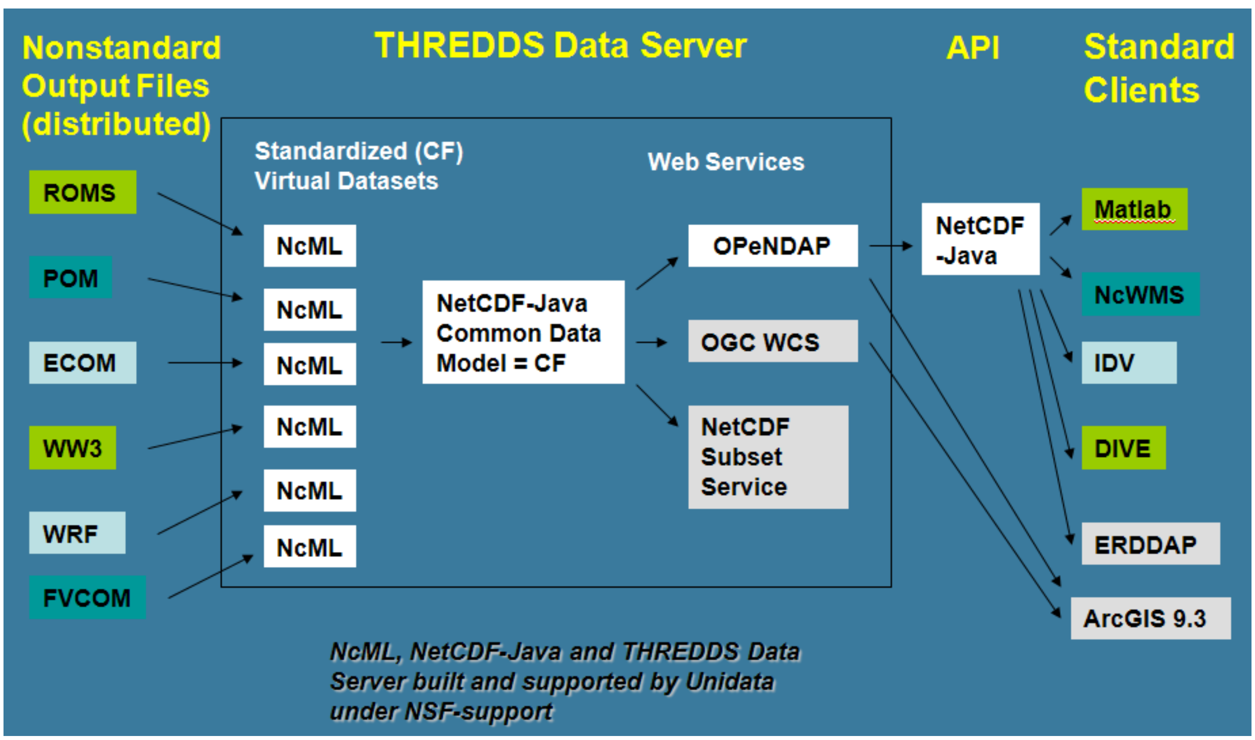
The NetCDF Markup Language (NcML) is an XML dialect that allows creation of Common Data Model (CDM) datasets. The NcML is an XML representation of netCDF metadata, and is similar to the netCDF CDL (network Common data form Description Language), except the XML syntax.

An NcML document is an XML document that uses NcML, and defines a CDM dataset. The purpose of NcML is to allow:

1. Metadata to be added, deleted, and changed.
2. Variables to be renamed, added, deleted and restructured.
3. Data from multiple CDM files to be combined.

NcML can be used to create a virtual CF-compliant NetCDF dataset, supplying or modifying the necessary metadata via XML, while accessing data from existing non-CF-compliant NetCDF, GRIB or HDF files as the following figure demonstrates.

Interoperability Experiment on Model Data in the Gulf Of Maine[4]



Ref: http://www.unidata.ucar.edu/software/netcdf/ncml/

http://stommel.tamu.edu/~baum/ncml.html

[4] Taken from Rich Signell's presentation at the IOOS DIF IPT Workshop; August 2009

1.4.6 Hierarchical Data Format (HDF)

The Hierarchical Data Format (HDF) is a self-defining file format for transfer of various types of data between different machines. The HDF library contains interfaces for storing and retrieving compressed or uncompressed raster images with palettes, and an interface for storing and retrieving N-dimensional scientific datasets together with information about the data, such as labels, units, formats, and scales for all dimensions. HDF can be tailored to specific data models including scientific data sets, satellite data, and point data.

HDF at present includes two data management formats, i.e. HDF4 and HDF5. Both HDF4 and HDF5 were designed to be a general scientific format, adaptable to virtually any scientific or engineering application, and also have been used successfully in non-technical areas. HDF5 addresses some of the deficiencies of HDF4, provides a richer data model, with emphasis on efficiency of access, parallel I/O, and support for high-performance computing. HDF5 is a general purpose library and file format for storing scientific data. HDF5 addresses some of the deficiencies of HDF 4.

HDF4 and HDF5 are both widely used in government, academia, and industry. There are more than 200 distinct applications of the formats and an estimated 1.6 million users of NASA data alone. It is also the base format for a number of community standards, such as HDF-EOS, the standard for NASA's Earth Observing System (EOS), and NeXus, the standard for Neutron, Xray and Muon Science.

HDF supports both C and FORTRAN interfaces, and it has been successfully ported to a wide variety of machine architectures and operating systems.

Ref: http://www.hdfgroup.org/

1.4.7 World Meteorological Organization (WMO) Formats

World Meteorological Organization (WMO) specifies a number of standard formats and data representation forms for exchange of data and products within and between the operational meteorological and hydrological communities. Use of these standards is mandatory for international exchange of operational meteorological data and products.

1.4.7.1 FM System of Numbering Codes

Coded messages are used for the international exchange of meteorological information comprising observational data provided by the WWW Global Observing System and processed data provided by the WWW Global Data-processing System. Coded messages are also used for the international exchange of observed and processed data required in specific applications of meteorology to various human activities and for exchanges of information related to meteorology.

The codes are composed of a set of code forms and binary codes made up of symbolic letters (or groups of letters) representing meteorological or, as the case may be, other geophysical elements. In messages, these symbolic letters (or groups of letters) are transcribed into figures indicating the value or the state of the elements described. Specifications have been defined for the various symbolic letters to permit their transcription into figures. In some cases, the specification of the symbolic letter is sufficient to permit a direct transcription into figures. In other cases, it requires the use of code figures, the specifications of which are given in code tables. Furthermore, a certain number of symbolic words and symbolic figure groups have been developed for use as code names, code words, symbolic prefixes or indicator groups.

Each code form bears a number, preceded by the letters FM. This number may be followed by a Roman numeral to identify the session of Commission for Synoptic Meteorology or Commission for Basic Systems which either approved the code form as a new one or made the latest amendment to its previous version.

Furthermore, an indicator term is used to designate the code form colloquially and is therefore called a "code name". In some cases, this code name is included as a symbolic prefix in the code form and during transmission ensures ready identification of the type of report (e.g. TESAC). The FM code forms, together with the corresponding code names:

WMO FM 12	SYNOP	Surface Synoptic Reports
WMO FM 13	SHIP	Ship Synoptic Reports
WMO FM 15	METAR	Aviation Routine Observations
WMO FM 16	SPECI	Special Aviation Weather Change
WMO FM 18	BUOY	Buoy Observations
WMO FM 20	RADOB	Radar Observations
WMO FM 22	RADREP	Radiological Data
WMO FM 32	PILOT	Upper Level Wind
WMO FM 33	PILOT SHIP	Upper Level Wind
WMO FM 34	PILOT MOBIL	Upper Level Wind
WMO FM 35	TEMP	Upper Level Observations
WMO FM 36	TEMP SHIP	Upper Level Observations
WMO FM 37	TEMP DROP	Aircraft Dropsonde Obs.
WMO FM 38	TEMP MOBIL	Upper Level Observations
WMO FM 39	ROCOB	Rocketsonde Reports
WMO FM 40	ROCOB SHIP	Rocketsonde Reports
WMO FM 41	CODAR	Aircraft Report
WMO FM 42	AMDAR	Aircraft Report
WMO FM 44	ICEAN	Ice report
WMO FM 45	IAC	Ship surface observation

WMO FM 46	IAC FLEET	Ship surface observation
WMO FM 47	GRID	Gridded Data
WMO FM 50	WINTEM	Upper-level Winds, Temperatures
WMO FM 51	TAF	Terminal Aerodrome Forecasts
WMO FM 53	ARFOR	Aviation Routine Forecasts
WMO FM 54	ROFOR	On-Route Aviation Forecasts
WMO FM 57	RADOF	Radiological Dose Predictions
WMO FM 61	MAFOR	Shipping Area Forecasts
WMO FM 62	TRACKOB	Oceanographic Data
WMO FM 63	BATHY	Oceanographic Data
WMO FM 64	TESAC	Oceanographic Data
WMO FM 65	WAVEOB	Oceanographic Data
WMO FM 67	HYDRA	Hydrological River Reports
WMO FM 68	HYFOR	Hydrological Forecast
WMO FM 71	CLIMAT	Surface climatic data
WMO FM 72	CLIMAT SHIP	Surface marine climatic data
WMO FM 73	NACLI, CLINP, SPCLI, CLISA, INCLI	Oceanic climatic data
WMO FM 75	CLIMAT TEMP	Upper-air climatic data
WMO FM 76	CLIMAT TEMP SHIP	Upper-air marine climatic data
WMO FM 81	SFAZI	Special Atmospheric Reports
WMO FM 82	SFLOC	Special Atmospheric Reports
WMO FM 85	SAREP	Satellite Cloud Interpretations
WMO FM 86	SATEM	Satellite Remote Upper Soundings
WMO FM 87	SARAD	Satellite Radiance Observations
WMO FM 88	SATOB	Satellite Temps & Radiance Balance
WMO FM 92	GRIB	Gridded Binary Data
WMO FM-94	BUFR	Binary Universal Form for the Representation of met. Data
WMO FM-95	CREX	Coded raw data table driven obs.

Ref: http://www.wmo.ch/pages/prog/www/WMOCodes/ManualCodesGuides.html

1.4.7.2 WMO FM 92 (GRIB)

Gridded Binary (GRIB) is a data format commonly used in meteorology to store historical and forecast weather data. It is standardized by the WMO's Commission for Basic Systems, and is described in WMO Manual on Codes No.306 under the number FM 92. GRIB superseded the Aeronautical Data Format (ADF).

GRIB is based on a mixed sequential/array data model, and a hardwired metadata model; it was primarily designed to exchange gridded data generated by numerical weather prediction models. It is an efficient method for transmitting and archiving large volumes of two-dimensional meteorological and oceanographic data and is widely used for storage and exchange of gridded data within the meteorological community. While it lacks a standard API, GRIB demonstrates very good data compression capability.

The description of the data is encoded in tables, which comprise a major part of the GIB documentation. The use of tables allows GRIB data to be exchanged without reliance on a single language (the tables are available in English, French, Spanish, Russian, Chinese and Arabic). As a consequence, GRIB data cannot be read without application of the accompanying tables, which are available in both printed and digital form.

Currently there are three versions of GRIB:

- version 0 was used to a limited extent by projects such as TOGA, and is no longer in operational use;
- version 1 (current sub-version is 2) is used operationally world-wide by most meteorological centers, for Numerical Weather Prediction output (NWP);
- version 2, also known as GRIB second edition, is rolling out, and data is changing over to this format.

Some of the second-generation GRIB are used for derived product distributed in Eumetcast of Meteosat Second Generation. Another example is the NAM (North American Mesoscale) model.

GRIB is the standard used by all of the operational meteorological centers (e.g. NCEP and the European Center for Medium-Range Weather Forecasts) for exchange of their gridded data and forecasts and is being used by NWS to distribute the gridded National Digital Forecast Database (NDFD).

The lack of the standard API required the user to write a special code to work with the GRIB2 library, and then refer to the WMO's specifications to decipher the metadata (e.g., variable type, variable unit, reference date time, valid date time, etc). To resolve this deficiency the NWS Meteorological Development Laboratory (MDL) created a driver for the GRIB2 library, known as "degrib" (aka "NDFD GRIB2 decoder"), which can greatly facilitate the work with GRIB data by:

- providing the ability to convert from GRIB2 to selected file formats such as ESRI shapefiles (.shp), ASCII comma sepearated files (.csv), NetCDF files, and binary float files (.flt) (useful in conjunction with GrADS, ESRI ArcGIS, or ESRI ArcView 3 + Spatial Analyst extension);
- enable users to understand the metadata produced by the GRIB2 library without needing to refer to the WMO's specifications by creating an ASCII (.txt) file that does the necessary lookups in the WMO's GRIB2 specification tables.

Ref: http://www.wmo.ch/pages/prog/www/WMOCodes/ManualCodesGuides.html

1.4.7.3 WMO FM 94 (BUFR)

The Binary Universal Form for the Representation of Meteorological Data (BUFR) is a binary format designed for the exchange of meteorological point data. BUFR files are stream-based and consist of a number of consecutive records. A BUFR record containing observational data of any sort also contains a coded description of what those data are: the description includes identifying the parameter in question (height, temperature, pressure, latitude, date and time), the units, any decimal scaling that may have been employed to change the precision from that of the original units. As with GRIB, this description is encoded in tables which comprise the major part of the BUFR documentation, with the consequence that BUFR data cannot be read without application of the appropriate tables. Furthermore, entries in the tables have only been defined for meteorological, marine and hydrological parameters.

Similar to GRIB, BUFR demonstrates excellent data compression capability, which is a significant advantage against HDF and netCDF. However, in the latest versions of HDF and netCDF the compression is catching up with BUFR. In addition, datasets encoded in BUFR can be converted into other formats, e.g. netCDF and HDF; a number of conversion tools have been developed.

Ref: http://www.wmo.ch/pages/prog/www/WMOCodes/ManualCodesGuides.html

1.4.7.4 WMO FM 95 (CREX)

The Character form for Representation and Exchange of data (CREX) is an alphanumeric table-driven code form designed in order to replace the various other alphanumeric formats, and to complement BUFR in areas where BUFR was not feasible, e.g. due to the lack of computer data handling capabilities. In many ways, CREX is very similar with BUFR: CREX messages contain metadata that tell what the actual data are and how they should be interpreted, data description section has to be included before actual data and managing CREX messages should require only minor changes from BUFR decoder. But unlike BUFR, CREX generated messages are human readable. CREX became operational in May 2000.

Ref: http://www.wmo.ch/pages/prog/www/WMOCodes/ManualCodesGuides.html

1.4.8 OPeNDAP

OPeNDAP, an acronym for "Open-source Project for a Network Data Access Protocol", is a data transport architecture, service and protocol widely used by governmental agencies to serve satellite, weather and other observed earth science data. The project is maintained by OPeNDAP, Inc, a publicly-funded non-profit organization that also provides free reference implementations of OPeNDAP protocols, servers and clients.

OPeNDAP service is based on a Data Access Protocol (DAP), a data transmission protocol designed specifically for science data. The protocol relies on the widely used and stable HTTP and MIME standards, and provides means to accommodate gridded data, relational data and time series, as well as allowing users to define their own data types. A certain DAP indifference to a data type gives OPeNDAP a positive advantage over OGC Web services, e.g. OPeNDAP can serve irregularly gridded data while WCS can work only with regular grids.

An OPeNDAP client could be an ordinary browser, although this gives limited functionality. Usually, an OPeNDAP client is a graphics program (like GrADS, Ferret or ncBrowse) or web application (like DChart) linked with an OPeNDAP library.

An OPeNDAP client sends requests to an OPeNDAP server, and receives various types of documents or binary data as a response. One such document is called a DDS (received when a DDS request is sent), that describes the structure of a data set. A data set, seen from the server side, may be a file, a collection of files or a database. Another document type that may be received is DAS, which gives attribute values on the fields described in the DDS. Binary data is received when the client sends a DODS request.

OPeNDAP provides software which makes local data accessible to remote locations regardless of local storage format. OPeNDAP also provides tools for transforming existing applications into OPeNDAP clients, i.e. enabling them to remotely access OPeNDAP served data.

An OPeNDAP server can serve an arbitrarily large collection of data. Data on the server is often in Hierarchical Data Format (HDF) or NetCDF format, but can be in any format including a user-defined format. Compared to ordinary file transfer protocols (e.g. FTP), a major advantage using OPeNDAP is the ability to retrieve subsets of files, and also the ability to aggregate data from several files in one transfer operation. Several different DAP servers have been developed and maintained, e.g. Hyrax; TDS; GDS; PyDAP; Dapper; FDS; CODAR.

Currently, the OPeNDAP development team is running a project that will provide automated capabilities to serve data from OPeNDAP via OGC Web service interfaces. However, the project meets some serious challenges. While the interaction with WCS and WMS is relatively easy as most data served by OPeNDAP is gridded and is semantically closest to the OGC coverage data model, the SOS specification creates serious challenges for OPeNDAP.

OPeNDAP has been commonly used throughout NOAA for serving of oceanographic, meteorological and climatological data. Recently, OPeNDAP software had demonstrated certain issue with security, and had temporarily been banned from use in federal resources; however, the problem was fixed in the subsequent releases, and the restriction was called off.

1.4.8.1 Live Access Server (LAS)

The Live Access Server (LAS) is a highly configurable Web server designed to provide flexible access to geo-referenced scientific data through the use of OPeNDAP networking. LAS is widely used by the educational and scientific communities as well as the government agencies and other institutions like NOAA, NASA, Navy, DoE, and several Multi-Agency Programs.

The LAS allows user to download and visualize data using a simple graphical user interface. Ferret is the default visualization application used by LAS, though other applications (Matlab, IDL, GrADS, ...) can also be used.

In addition, LAS enables the web user to:

- visualize data with on-the-fly graphics
- request custom subsets of variables in a choice of file formats
- access background reference material about the data (metadata)
- compare (difference) variables from distributed locations

For the data provider, LAS allows to:

- unify access to multiple types of data in a single interface
- create thematic data servers from distributed data sources
- offer derived products on the fly
- remedy metadata inadequacies (poorly self-describing data)
- offer unique products (e.g. visualization styles specialized for the data)

1.4.8.2 THREDDS

THREDDS (Thematic Real-time Environmental Distributed Data Services) is a middleware to bridge the gap between data providers and data users. The current focus of THREDDS development is the THREDDS Data Server (TDS), which actually serves the contents of the datasets, in addition to providing catalogs and metadata for them.

The TDS uses the Unidata CDM to read datasets in various formats, and serves them through OPeNDAP, OGC WCS, NetCDF subset, and bulk HTTP file transfer services. The first three allow the user to obtain subsets of the data, which is crucial for large datasets.

The TDS aggregates many files into virtual datasets, which insulates users from the details of file storage and naming, and greatly simplifies user access to large collections of files. The TDS is open source Java application, and runs inside the popular Tomcat Servlet container.

Ref: http://opendap.org/

http://www.unidata.ucar.edu/projects/THREDDS/

1.5 Information Architecture Frameworks

1.5.1 Federal Enterprise Architecture (FEA)

The Federal Enterprise Architecture (FEA) is a business and performance-based framework for government-wide improvement under collaborative development by federal agencies, the Federal Chief

Information Officers (CIO) Council and the Office of Management and Budget (OMB). The FEA includes a collection of five inter-related reference models:

- **Business Reference Model (BRM)** provides a means to describe the business operations of the federal government independent of the agencies that conduct the business.
- **Data Reference Model (DRM)** addresses the goals of improving effectiveness of IT investments and sharing data by promoting characterization, exchange, and documentation of data across agencies. The DRM exists only as a framework, whereas the other models consist of components.
- **Service Component Reference Model (SRM)** categorizes components (self contained process, service, or IT capability with pre-determined functionality that may be exposed through a business or technology interface) with respect to how they support business and/or performance objectives.
- **Technical Reference Model (TRM)** identifies the technology components that collectively support the adoption and implementation of technical architectures, and provides the foundation to advance the re-use of technology and component services across the Federal Government through standardization.
- **Performance Reference Model (PRM)** helps agencies measure the performance of major IT investments and their contribution to agency performance.

Federal government mandates each Federal agency to plan, procure, and deliver business products and services through their Enterprise Architectures. Federal agencies use FEA reference models to set strategic goals and plan and develop annual budgets that describe to OMB how investments "align" to the business, performance, service component, and technical reference models. Agencies describe their IT investments in terms of the business operations supported, functional capabilities delivered, technologies used to build or deliver the capabilities, and performance results. The FEA provides a common language to describe the relationship among these areas to reduce redundancy, facilitate information sharing, focus on citizens and customers, and maximize IT investments to achieve agency missions.

Ref: http://www.whitehouse.gov/omb/e-gov/fea/

1.5.1.1 FEA Geospatial Profile

The Geospatial Profile of the Federal Enterprise Architecture (FEA) developed by the CIO Council and Federal Geographic Data Committee (FGDC) aims to help agencies to better understand how geospatial data can be essential to their missions.

The purpose of the Geospatial Profile is to provide enterprise architects with means to assess the role of geospatial resources in Federal agencies activity, and incorporate the resources into their business and IT operations, making the agencies "geo-enabled". A "geo-enabled" process, whether from a business or technology perspective, is one that generates, uses, or displays digital geospatial data. A "geo-enabled" organization optimizes the ability for the agency to use geospatial resources.

The Geospatial Profile uses the five FEA reference models as a framework for the approaches outlined as they are the overarching structure by which Federal agency designs its enterprise architecture and formulates information technology (IT) investment strategies. This Profile, like other FEA Profiles, provides guidance on how to incorporate this cross-cutting discipline in the context of the FEA reference models and the many FEA Lines of Business where it may apply.

Ref: http://www.cio.gov/library/library.cfm

1.5.2 NOAA Technical Reference Model (TRM)

NOAA Technical Reference Model (TRM) is a technology layer of the NOAA Enterprise Architecture. The NOAA TRM organizes and identifies NOAA's enterprise IT standards, specifications and products approved for enterprise-wide use.

The NOAA TRM provides an authoritative source of technologies that have been vetted at the enterprise level and determined to be consistent with NOAA's overall business and technology drivers. Additionally, the structure (e.g., taxonomy) and content (e.g., standards profiles) of the TRM provides a common language to describe technologies in a consistent fashion, thereby enabling discovery and reuse of these technologies across NOAA.

The NOAA TRM categorizes standards in one of three categories for implementation purposes:

- **Mandatory** – compliance or an approved waiver is required.
- **Recommended** – implementation of standards in this category is encouraged but not absolutely required.
- **Prohibited** – denotes those technologies which should not be used under any circumstances due to major security issues associated with them or other compelling concerns.

The NOAA TRM taxonomy is hierarchical and reflects a blend of the Federal Enterprise Architecture TRM and Department of Commerce TRM and Standards Profile taxonomies, modified to fit NOAA's unique requirements:

- **Service Area** – the highest level grouping of technology topics; the level is highly abstract, and is included primarily to facilitate mapping of the NOAA TRM to the FEA TRM.
- **Service Category** – a further decomposition of the high-level service area into a logical grouping of more granular technology topics.
- **Service Standard** – identification of the specific IT standards, specifications or products that have been approved (or prohibited) for use within NOAA.

Ref: http://www.cio.noaa.gov/IT_Groups/noaa_cio_EA_Committee.html

1.5.3 NASA Enterprise Architecture (EA)

NASA Enterprise Architecture (EA) is based on the Federal Enterprise Architecture and the associated supporting reference models. NASA has extended the Federal framework and reference models, where appropriate, utilizing commercial best practices, to capture the unique elements of NASA science and research and technology missions. NASA's Enterprise Architecture Review process was recognized as a federal Best Practice, and NASA EA Team received the E-Gov Institute's "Government Civilian Leadership in Government Transformation Award" in 2006.

The NASA EA focuses on leveraging the investments in legacy systems and driving the design on the emerging systems. The EA leverages existing systems that NASA has in place, built separately by Centers and Programs over several decades. The NASA EA and the associate reference models mold these systems into an integrated or federated infrastructure aligned with the Agency's mission and business needs.

The NASA EA provides a mission-driven approach to designing and implementing, partnering, or procuring new information technology systems and services. NASA EA encompasses 5 segments: one for each of four Mission Directorates, which are NASA's Lines of Business (LoB), and one for agency cross-cutting capabilities (IT, CFO).

The NASA EA is structured into three major investment categories:

- **Office Automation, IT Infrastructure, and Telecommunications IT (OAIT)**
  This category includes Office Automation investments that provide general purpose computing (e.g. email, desktops, help desk services), regardless of the program or project supported or fund source. Nine portfolios (Voice, WAN, LAN, Video, Desktop Hardware/Software, Data Centers, Application Services, Messaging and Collaboration, and Public Web) have been defined across three major service areas (Communications, Computing, and Electronic Work Environment.)

- **Multi-Program/Project IT**
  Multi-Program/Project IT is defined as IT infrastructure, products, and services that are not part of OAIT but do meet IT requirements that are not unique to a single program/project. Three major service areas and nine portfolios have also been defined for this category. The service area names are identical to those in the OAIT category, as are the portfolio names with one exception: Compute Engine Hardware/Software replaces the Desktop Hardware/Software portfolio. This is in recognition that in this investment category, computing platforms may range from science and engineering workstations to supercomputers.

- **Program Unique IT**
  Program Unique IT is defined as infrastructure, products and services that are either physically embedded in a flight or test article, or exist solely to meet the requirements of a single specific program or project. The portfolio structure for this category is based on the NASA themes and programs.

Ref: http://www.spaceref.com/news/viewsr.html?pid=19188

1.5.4 The Open Group Architecture Framework (TOGAF)

The Open Group Architecture Framework (TOGAF) is a comprehensive approach to the design, planning, implementation, and governance of an enterprise information architecture defines how to enterprise information architectures for a wide range of applications. TOGAF is developed and maintained by the Architecture Forum of The Open Group. It was originally developed in the mid-1990's, and has continuously evolved since then.

TOGAF is broader in scope than its defense counterparts, DoDAF and MODAF; it organizes architectures into four domain levels:

- **Business Architecture** – defines business strategy, governance, organization, and key business processes;
- **Application Architecture** – specifies individual application systems to be deployed;
- **Data Architecture** – defines structure of an organization's logical and physical data assets and associated data management resources; and
- **Technology Architecture** – specifies software infrastructure intended to support the deployment of core, mission-critical applications

TOGAF is an industry standard architecture framework that may be used freely by any organization wishing to develop information systems architecture for use within that organization.

Ref: http://www.opengroup.org/togaf/

1.5.5 Department of Defense Architecture Framework (DoDAF)

The Department of Defense Architecture Framework (DoDAF) specifies the enterprise architecture for U.S. Department of Defense (DoD) applications. DoDAF is well suited to large systems and systems-of-systems (SoS) with complex integration and interoperability issues. Although DoDAF is primarily focused on defense applications, it can also be applied to commercial systems.

DoDAF is administered by the U. S. Undersecretary of Defense for Business Transformation's DoDAF Working Group. It was formerly named C4ISR (Command, Control, Communications, Computers, Intelligence, Surveillance and Reconnaissance) Architecture Framework. Other derivative frameworks based on DoDAF include the Ministry of Defence (United Kingdom) Architecture Framework (MODAF), and the NATO Architecture Framework (NAF).

DoDAF organizes enterprise architectures into four basic view sets:

- All View (AV) with two work products;
- Operational View (OV) with seven work products;
- Systems View (SV) with 11 work products;
- Technical Standards View (TV) with two work products.

In practice, most systems only use a subset of the DoDAF view products to specify their system architectures. Like other architecture framework approaches, DoDAF defines a data repository for holding its work products. The DoDAF repository is called the Core Architecture Data Model (CADM).

The most recent version of the DoDAF specification is DoDAF v2.0, which was published in 2009, and consists of three volumes:

- Volume I (Manager's Guide – Introduction, Overview and Concepts) introduces DoD architecture concepts and provides general guidance for development, use, and management of DoD architectures.
- Volume II (Architect's Guide – Architectural Data and Models) describes the Meta-model data groups, and their associated models from a technical viewpoint.
- Volume III (Developer's Guide – DoDAF Meta-model Physical Exchange Specification) describes the Data Models' structure, relationships, associations, and business rules to introduce the constructs needed to enable exchange of data and derived information among users..

Ref: http://www.defenselink.mil/cio-nii/policy/eas.shtml

1.5.6 National Information Exchange Model (NIEM)

The National Information Exchange Model (NIEM) is a partnership of the U.S. Department of Justice and the Department of Homeland Security. NIEM is a framework to provide a foundation for seamless local, state, tribal, and federal interagency information exchange.

NIEM enables information sharing, focusing on information exchanged among organizations as part of their current or intended business practices. The NIEM exchange development methodology results in a common semantic understanding among participating organizations and data formatted in a semantically consistent manner. NIEM will standardize content (actual data exchange standards), provide tools, and managed processes.

NIEM relies on the Global Justice XML Data Model (GJXDM) to facilitate timely and secure information sharing across the whole of the justice, public safety, emergency and disaster management, intelligence, and homeland security domains.

Ref: http://www.niem.gov/library.php

1.5.7 Information Sharing Environment Enterprise Architecture Framework (ISE EAF)

The Information Sharing Environment (ISE) was developed in order to provide the means for sharing terrorism information among all appropriate Federal, State, local and tribal entities, foreign partners, and the private sector through the use of policy guidance and technologies.

The ISE Enterprise Architecture Framework (ISE EAF) was developed as a common framework to facilitate the secure connection to ISE, and sharing information across the ISE. The ISE EAF is a part of the Architecture Program of the ISE, which also includes ISE Drivers and Requirements Specification, and ISE Profile and Architecture Implementation Strategy (ISE PAIS).

The ISE Drivers and Requirements Specification describes the authoritative mandates (e.g., Executive Orders, Public Laws) that direct the ISE. These ISE Drivers and Requirements are strategic in nature and establish direction to bring about ISE specific results.

The ISE PAIS identifies implementation guidance compiled from the ISE Drivers and Requirements Specification. The ISE PAIS provides guidance to ISE participants as they seek to implement information sharing capabilities, connect to other ISE participants, expose data, and access ISE data and services.

The ISE EAF provides a common architectural structure for agencies to incorporate their information sharing capabilities into the ISE. The ISE EAF provides a logical structure of ISE business processes, information flows and relationships, services, and high-level data packet descriptions and exchange relationships.

Ref: http://www.ise.gov/pages/eaf.html

1.5.8 NextGen EA

The NextGen EA consists of the EA framework, the Community Model, and the NextGen activities compiled into an Enterprise Segment Activity Inventory (ESAI).

The NextGen Community Model provides a highly-visual view of the significant concepts, actors, roles, and the relationships among them, within the context of the NextGen environment. It acts as the foundation for other NextGen EA work products and serves to promote NextGen stakeholder involvement and understanding. It forms the basis for capturing, understanding, and analyzing the needs of users in the context of the activities they perform and the enterprise services provided to them, and allows NextGen stakeholders to see how they fit into the community as a whole.

The Enterprise Segment Activity Inventory (ESAI) summarizes and compiles the relevant aspects of the NextGen EA from the point of an operational activity. Activities are defined as the operational work that users (or systems) perform to accomplish specific goals and objectives within the context of the NextGen environment.

The NextGen EA framework relies on integration of the FEA Framework and the DoDAF. The integration of these two architecture frameworks allows addressing the broader stakeholder community, which is important since NextGen EA is intended to be a collaboration tool across multiple government agencies and the private sector, and coordinate mission objectives and funding priorities among them. The combined framework establishes a common lexicon and defines a structure for organizing information describing the scope of the architecture and how the architectural layers are related to each other.

Ref: http://www.jpdo.gov/

1.6 Security Standards

1.6.1 Federal Enterprise Architecture Security and Privacy Profile (FEA SPP)

The Federal Enterprise Architecture Security and Privacy Profile (FEA SPP) was developed by the Office of Management and Budget and the Federal Chief Information Officers Counci. Assessments of policies and guidance related to enterprise architecture, security, privacy, and capital planning documents ensured that the FEA SPP is relevant and complementary to current activities. The FEA SPP was field tested through validation exercises at the Department of Housing and Urban Development and the Department of Justice. During these exercises, senior cross-functional teams walked through the FEA

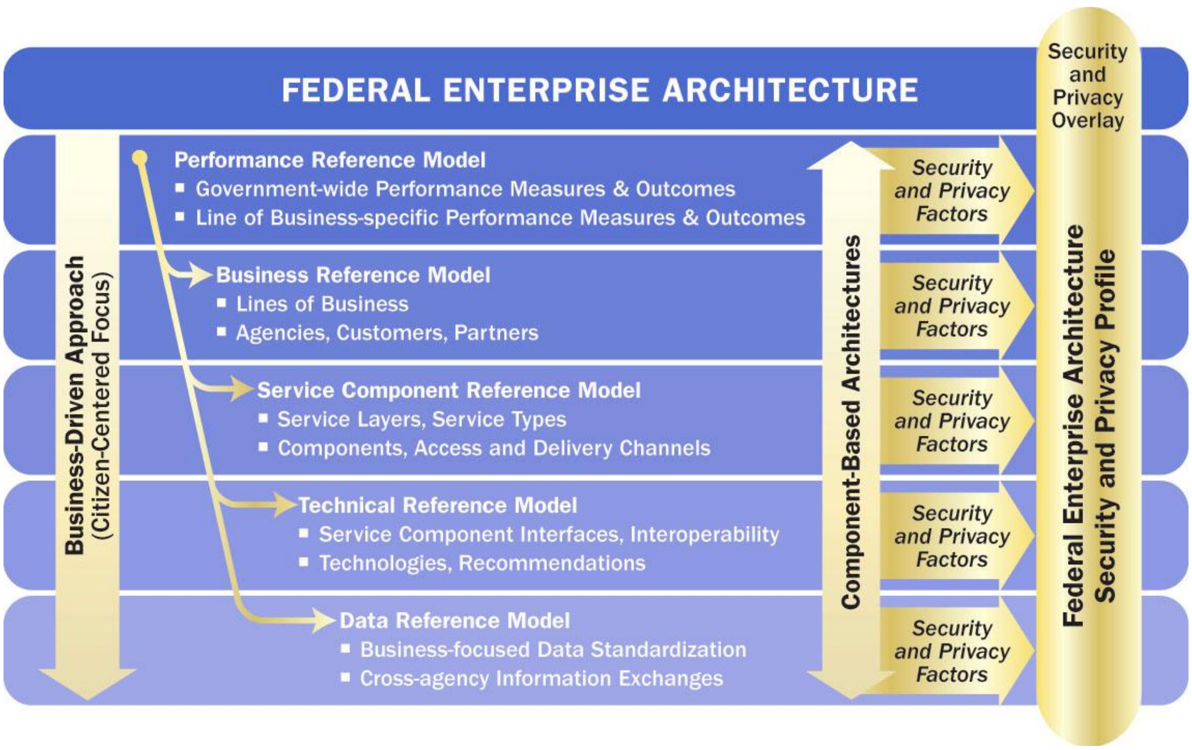
SPP methodology to ensure its usability and applicability. Although limited in duration and scope, these exercises led to meaningful changes in the business processes.

In short, the SPP:

- Promotes an understanding of the organization's security and privacy requirements, its capabilities to meet those requirements and the risks to its business;
- Helps program executives select the best way to meet the requirements and improve current capabilities, using standards and services that are common to the enterprise or government;
- Improves agencies' processes for incorporating privacy and security into major investments.

The FEA SPP evaluates enterprise-level security and privacy in the context of the Federal Enterprise Architecture (FEA).The profile cuts across all five layers of the FEA' business, service component, performance, technical and data reference models as represented in the figure below.

Federal Enterprise Architecture and FEA SPP[5]



The profile outlines its methodology that asks agencies to:

- Identify the program's needs and capabilities;
- Analyze how to effectively address those needs with a consideration to using existing systems to reduce costs;

[5] Taken from The Federal Enterprise Architecture Security and Privacy Profile

- Select the tools to improve the security and privacy of system including ensuring the agency has asked for adequate funding and the effort is coordinated across the department.

The profile outlines 17 security and 17 privacy control areas, which provide a common terminology and framework. The security control areas includes risk assessment, planning, system and services acquisition, while the privacy control areas include policies and procedures, monitoring and measuring and acceptable use.

The FEA SPP does not replace or alter a wide range of information security standards and guidance provided by National Institute of Standards and Technology (NIST); it does seek to capture the outputs of system-level security activities and use them to support enterprise decisions.

There is no equivalently rich source of system-level privacy guidance, but the FEA SPP does consider existing guidance concerning system-level privacy activities to support enterprise decision making. It also seeks to add depth to the privacy discussion so that it can be treated equivalently to security.

Ref: http://www.cio.gov/library/library.cfm

http://govitwiki.com/wiki/IT_Security

1.6.2 NIST Risk Management Framework, Standards and Special Publications (SP)

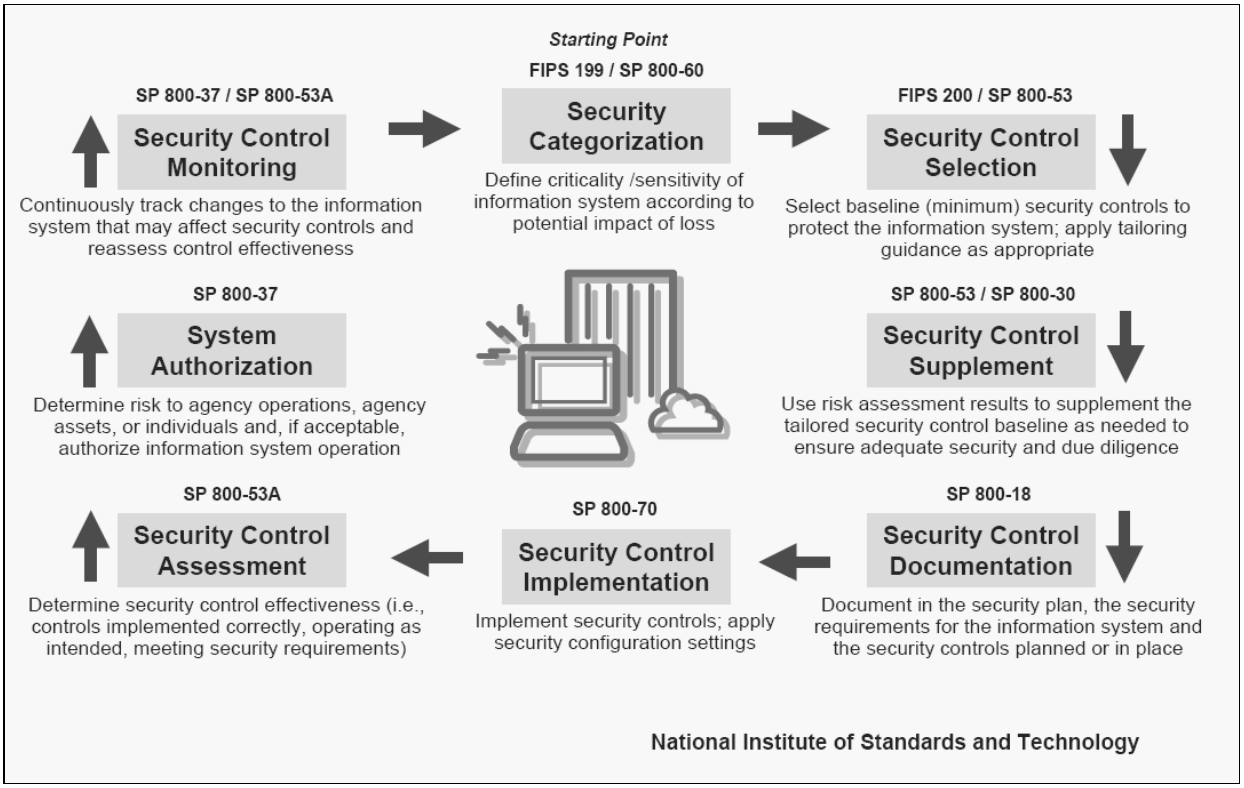
Management of risks involves the risks to the agency with the operation of an information system or information security management system. Risk management is an effective frame work for selecting appropriate security controls for an information system and assist in selecting of appropriate security controls to protect assets.

NIST have suggested a Risk Management Framework that defines key activities in managing enterprise-level risk, i.e. risk resulting from the operation of an information system (Figure 1.1.7); the activities can be applied to both new and legacy information systems within the context of the system development life cycle and the Federal Enterprise Architecture:

1. **Categorize** the information system and the information resident within that system based on impact. FIPS 199 and NIST SP 800-60 Revision 1 (Volume 1, Volume 2)
2. **Select** an initial set of security controls for the information system based on the FIPS 199 security categorization and the minimum security requirements defined in FIPS 200; apply tailoring guidance as appropriate; and supplement the tailored baseline security controls based on an assessment of risk and local conditions including organization-specific security requirements, specific threat information, cost-benefit analyses, or special circumstances. FIPS 200 and NIST SP 800-53, Revision 3
3. **Implement** the security controls in the information system.
4. **Assess** the security controls using appropriate methods and procedures to determine the extent to which the controls are implemented correctly, operating as intended, and producing the desired outcome with respect to meeting the security requirements for the system. NIST SP 800-53A

5. **Authorize** information system operation based upon a determination of the risk to organizational operations, organizational assets, or to individuals resulting from the operation of the information system and the decision that this risk is acceptable. NIST SP 800-37
6. **Monitor** and assess selected security controls in the information system on a continuous basis including documenting changes to the system, conducting security impact analyses of the associated changes, and reporting the security status of the system to appropriate organizational officials on a regular basis. NIST SP 800-37 and SP 800-53A

NIST Risk Management Framework / Security Life Cycle



1.6.2.1 FIPS 199

Federal Information Processing Standard 199 defines "Standards for Security Categorization of Federal Information and Information Systems", and provides security categorization guidelines for information and information systems. Security categorization make available a common framework and method for expressing security.

The security categories are based on the potential impact of certain events, which jeopardize the information and information systems needed by the organization to accomplish its assigned mission, protect its assets, fulfill its legal responsibilities, maintain its day-to-day functions, and protect individuals. Security categories are therefore to be used in conjunction with vulnerability and threat information while assessing the risk to an organization.

FIPS 199 defines three levels of potential impact on organizations or individuals in case of a breach of security (i.e., a loss of confidentiality, integrity, or availability); they are summarized in the table below:

Potential Impact Definitions for Security Objectives (FIPS 199 Publication)

Security Objective	Potential Impact: LOW	Potential Impact: MODERATE	Potential Impact: HIGH
Confidentiality Preserving authorized restrictions on information access and disclosure, including means for protecting personal privacy and proprietary information. [*44 U.S.C., SEC. 3542*]	The unauthorized disclosure of information could be expected to have a **limited** adverse effect on organizational operations, organizational assets, or individuals.	The unauthorized disclosure of information could be expected to have a **serious** adverse effect on organizational operations, organizational assets, or individuals.	The unauthorized disclosure of information could be expected to have a **severe or catastrophic** adverse effect on organizational operations, organizational assets, or individuals.
Integrity Guarding against improper information modification or destruction, and includes ensuring information non-repudiation and authenticity. [*44 U.S.C., SEC. 3542*]	The unauthorized modification or destruction of information could be expected to have a **limited** adverse effect on organizational operations, organizational assets, or individuals.	The unauthorized modification or destruction of information could be expected to have a **serious** adverse effect on organizational operations, organizational assets, or individuals.	The unauthorized modification or destruction of information could be expected to have a **severe or catastrophic** adverse effect on organizational operations, organizational assets, or individuals.
Availability Ensuring timely and reliable access to and use of information. [*44 U.S.C., SEC. 3542*]	The disruption of access to or use of information or an information system could be expected to have a **limited** adverse effect on organizational operations, organizational assets, or individuals.	The disruption of access to or use of information or an information system could be expected to have a **serious** adverse effect on organizational operations, organizational assets, or individuals.	The disruption of access to or use of information or an information system could be expected to have a **severe or catastrophic** adverse effect on organizational operations, organizational assets, or individuals.

The overall security category (SC) of an information type (IT) or information system (IS) can be expressed in the following general format:

$\mathbf{SC}_{\text{IT/IS}}$ = {(**confidentiality**, *impact*), (**integrity**, *impact*), (**availability**, *impact*)},

where the acceptable values for potential impact are LOW, MODERATE, HIGH, or NOT APPLICABLE.

Ref: http://csrc.nist.gov/publications/PubsFIPS.html

1.6.2.2 NIST SP 800-53

The purpose of the NIST Special Publication (SP) 800-53 is to provide guidelines for selecting and specifying security controls for information systems supporting the executive agencies of the federal government. The guidelines apply to all components of an information system.

Before selecting the controls from NIST 800-53, agencies need to have complete inventory of the assets involved in the scope. Assets involved in the scope would require a comprehensive risk assessment to determine the sensitivity/criticality of these assets. Depending on the categorization of these assets will determine an appropriate control from standard to mitigate relevant risk. In some cases supplemental controls may be required.

NIST 800-53 suggests 163 high level controls and 154 medium level controls. While NIST SP 800-53 is required for federal unclassified information system, NIST encourages its use in commercial space as well. Commercial organizations can utilize the NIST standard to create their security program, which will provide a road map to their security strategy and assist in making informed decisions for securing their information assets.

1.6.2.3 NIST SP 800-95

Although the NIST SP 800-95 is not a part of the Risk Management Framework, it is indispensable for the SOA. The flexibility of service-oriented architecture has a cost – it brings some additional concerns over security due to the number of communities, domains, and platforms that may be crossed in executing a business process based on SOA.

The NIST Special Publication 800-95, "Guide to Secure Web Services", addresses some specific security issues:

- confidentiality and integrity of data that is transmitted via Web services protocols in service-to-service transactions, including data that traverses intermediary services;
- functional integrity of the Web services that requires the establishment of trust between services on a transaction-by-transaction basis;
- availability in the face of denial of service attacks that exploit vulnerabilities unique to Web service technologies, especially targeting core services, such as discovery service, on which other services rely;
- SOAs are dynamic and can seldom be fully constrained to the physical boundaries of a single network;
- access protocol messages as well as data are usually transmitted over HTTP, which is allowed to flow without restriction through most firewalls;
- Transport Layer Security (TLS), which is used to authenticate and encrypt Web-based messages, cannot accommodate Web services' inherent ability to forward messages to multiple other Web services simultaneously.

On the other hand, Web services rely on existing protocols and will coexist with other network applications in the same environment. Therefore, the majority of Web service security problems can be mitigated by using traditional security tools, such as firewalls, intrusion detection systems (IDS), and secured operating systems, which have been in effect before implementation of Web services applications, along with adhering to the strong IT security policy.

The NIST SP 800-95 recommends enforcing the following measures to improve the security of Web services transactions:

- authentication and identity management across domains and environments;
- authorization and confidentiality (access control);
- integrity (no inappropriate modifications are made);
- availability (reliable service, no denial of service);
- non-repudiation (positive identification and inability to deny providing or receiving services);
- auditing and monitoring;
- security administration and policy management.

1.6.2.4 NIST SP 800-47

The NIST SP 800-47 “Security Guide for Interconnecting It Systems” provides guidance for planning, establishing, maintaining, and terminating interconnections between information technology (IT) systems that are owned and operated by different organizations, including organizations within a single federal agency.

The NIST 800-47 provides not a mandatory but rather a recommended approach for interconnecting IT systems. It provides a logical framework for those organizations that have not previously interconnected IT systems, and it provides information that other organizations may use to enhance the security of existing interconnections.

A system interconnection is defined as the direct connection of two or more IT systems for the purpose of sharing data and other information resources. The Publication describes various benefits of interconnecting IT systems, identifies the basic components of an interconnection, identifies methods and levels of interconnectivity, and discusses potential security risks associated with an interconnection.

The NIST SP 800-47 introduces a “life-cycle management” approach for interconnecting IT systems, with an emphasis on security. The four phases of the interconnection life cycle are addressed:

- planning the interconnection;
- establishing the interconnection;
- maintaining the interconnection;
- disconnecting the interconnection.

The Publication provides recommended steps for completing each phase, emphasizing security measures that should be taken to protect the connected systems and shared data.

In addition, NIST SP 800-47 contains guides and samples for developing an Interconnection Security Agreement (ISA) and a Memorandum of Understanding/Agreement (MOU/A). The ISA specifies the technical and security requirements of the interconnection, and the MOU/A defines the responsibilities of the participating organizations.

Finally, the Publication provides a guide for developing a System Interconnection Implementation Plan, which defines the process for establishing the interconnection, including scheduling and costs.

Ref: http://csrc.nist.gov/publications/PubsSPs.html

1.6.3 Security Assertion Markup Language (SAML)

Security Assertion Markup Language (SAML), developed by the OASIS Security Services Technical Committee, is an XML-based standard for exchanging authentication and authorization data between security domains, i.e. between an identity provider (a producer of assertions) and a service provider (a consumer of assertions).

SAML allows business entities to make assertions regarding the identity, attributes, and entitlements of a subject (an entity that is often a human user) to other entities, such as a partner company or another enterprise application. SAML ensure loose coupling at the identity management layer by standardizing mechanisms and formats for the communication of identity information between the domains, where the standard provides the insulating buffer.

There are four 'drivers' behind the creation of the SAML standard:

- *Limitations of Browser cookies*: Most existing Single-Sign On products use browser cookies to maintain state so that re-authentication is not required. Browser cookies are not transferred between DNS domains. So, a cookie obtained from www.abc.com, will not be sent in any HTTP messages to www.xyz.com. This may happen even within an organization that has separate DNS domains. Therefore, to solve the Cross-Domain SSO (CDSSO) problem requires the application of different technology. SAML provides the means to solve the CDSSO problem in a standardized orderly fashion.
- *SSO Interoperability*: Various products implement proprietary SSO and CDSSO. SAML allows users to obtain the same SSO product in all the domains.
- *Web Services*: Security within Web Services is still being defined. Most of the focus has been on how to provide confidentiality and authentication/integrity services on an end-to-end basis. The SAML standard provides the means by which authentication and authorization assertions can exchanged between communicating parties.
- *Federation*: The need to simplify identity management across organizational boundaries, allowing users to consolidate many local identities into a single (or at least a reduced set) Federated Identity..." [from the Security Assertion Markup Language (SAML) 2.0 Technical Overview].

Ref: http://saml.xml.org/

1.6.4 WS-Security

WS-Security (Web Services Security) is a communications protocol providing a means for applying security to Web services. Originally developed by IBM, Microsoft, and VeriSign, the protocol is now officially called WSS and developed via committee in Oasis-Open. Oasis-Open released WS-Security 1.0 April 19, 2004. Oasis-Open released version 1.1 on February 17, 2006.

The protocol contains specifications on how integrity and confidentiality can be enforced on Web services messaging. The WSS protocol includes details on the use of SAML and Kerberos, and certificate formats such as X.509.

WS-Security describes how to attach signatures and encryption headers to SOAP messages. In addition, it describes how to attach security tokens, including binary security tokens such as X.509 certificates and Kerberos tickets, to messages.

WS-Security incorporates security features in the header of a SOAP message, working in the application layer. Thus it ensures end-to-end security.

The following specifications are associated with WS-Security:

- WS-SecureConversation
- WS-Federation
- WS-Authorization
- WS-Policy
- WS-Trust
- WS-Privacy

Ref: http://www.ibm.com/developerworks/library/specification/ws-secure/

http://www.ibm.com/developerworks/webservices/library/specification/ws-secon/

http://www.ibm.com/developerworks/library/specification/ws-trust/

http://www.soaspecs.com/ws-security.php

http://www.w3.org/2002/ws/policy/

1.6.4.1 WS-SecureConversation

WS-SecureConversation is a Web Services specification that works in conjunction with WS-Security, WS-Trust and WS-Policy to allow sharing of security contexts. The purpose of WS-SecureConversation is to allow secure conversations between sites using Web Services for communication.

1.6.4.2 WS-Federation

WS-Federation specification is a part of the WS-Security framework, and defines mechanisms for allowing to broker information on identities, identity attributes and authentication. WS-Federation has been developed by a group of companies such as BEA Systems, BMC Software, CA, Inc., IBM, Layer 7 Technologies, Microsoft, Novell, Ping Identity, and VeriSign.

1.6.4.3 WS-Authorization

WS-Authorization is a specification for authorization data and policy management for Web services. It provides the mechanism to process attributes and protect resource access based on access policy. It allows for authorization policy to be configured and enforced at various levels of. It also provides client side authorization to allow clients to authorize the services they access.

1.6.4.4 WS-Policy

WS-Policy is a specification that allows Web services to use XML to advertise their policies (on security, QoS, etc.) and for Web service users to specify their policy requirements. Since September 2007, the WS-Policy has been a W3C recommendation.

WS-Policy represents a set of specifications that describe the capabilities and constraints of the security (and other business) policies on intermediaries and end points (for example, required security tokens, supported encryption algorithms, and privacy rules) and how to associate policies with services and end points.

1.6.4.5 WS-Trust

Web Services Trust Language (WS-Trust) is a specification that provides extensions to WS-Security, specifically dealing with the issuing, renewing, and validating of security tokens, as well as with ways to establish, assess the presence of, and broker trust relationships between participants in a secure message exchange. Using the extensions defined in WS-Trust, applications can engage in secure communication designed to work within the Web services framework.

WS-Trust defines a number of new elements, concepts and artifacts in support of that goal, including:

- the concept of a Security Token Service (STS) - a web service that issues security tokens as defined in the WS-Security specification.
- the formats of the messages used to request security tokens and the responses to those messages.
- mechanisms for key exchange

The WS-Trust specification was authored by representatives of a number of companies, and was approved by OASIS as a standard in March 2007.

1.6.4.6 WS-Privacy

WS-Privacy specification describes Web services and requesters privacy preferences and organizational privacy practices.

1.7 Compression Standards

1.7.1 Image Compression

1.7.1.1 JPEG-2000

JPEG-2000 is a wavelet-based image compression standard and coding system created by the Joint Photographic Experts Group committee to replace the discrete cosine transform-based JPEG standard. JPEG-2000 has also been published as an ISO standard, ISO/IEC 15444.

JPEG-2000 demonstrates a long list of features making it a superior compression format:

- Progressive transmission by pixel accuracy allows images to be reconstructed with different resolutions and pixel accuracy, as needed or desired, for different target devices; the image architecture of JPEG-2000 provides for efficient delivery of image data in Internet and client/server applications.
- Built-in error resilience tools, which allow for error detections and concealment, guaranteeing more reliable, better image transmission in noisy environments; the preceding JPEG standard has provision for restart intervals, but image quality suffers dramatically when bit errors are encountered.
- Region-Of-Interest (ROI) coding feature so that certain areas of an image can be coded with better quality than the rest of the image (background); in addition, the ROI area is placed at the beginning of the file's bitstream so that the ROI will be decompressed before the rest of the image, allowing more reliable, faster access to regions of an image that are deemed more important.
- Inclusion of metadata & ICC profiles provides a mechanism by which metadata, such as tonescale, colorspace, intellectual property rights or other file information can be included in the image file.
- Enhancement of compound documents compression; the preceding JPEG standard is rarely used in the compression of compound documents because of its poor performance when applied to bi-level (text) imagery.
- Low bit-rate compression performance enhancement allowing better compression at low bit rates (e.g. below 0.25 bpp) for highly detailed images; the preceding JPEG offered excellent rate-distortion performance in the mid and high bit-rates, but at low bit-rates the distortion are unacceptable.
- Lossless and lossy compression in a single codestream.
- Seamless compression of large images; the preceding JPEG image compression algorithm does not allow for images greater then 64K by 64K without tiling.
- Single decompression architecture to enhance interchange between applications; the preceding JPEG standard had over 40 modes, many of which are application specific and not used by the majority of the JPEG decoders.
- Improved support of computer generated imagery; the preceding JPEG standard was optimized for natural imagery and does not perform well on computer generated imagery.
- Enhanced support for alternate color spaces.

Ref: http://www.jpeg.org/jpeg2000/index.html

1.7.2 XML Compression

1.7.2.1 EXI

Efficient XML Interchange (EXI) is a W3C standard for compacting XML code. The EXI processor produces a very compact, high performance binary XML representation that was designed to work well for a broad range of applications. It simultaneously improves performance and significantly reduces bandwidth requirements without compromising efficient use of other resources.

EXI uses a grammar-driven approach that achieves very efficient encodings using a straightforward encoding algorithm and a small set of data types. As a result, EXI processors are simple and can be implemented on devices with limited capacity. EXI allows to compress XML and preserve the "streamability" of the XML stream, which makes possible, for example, to use a decoder to receive an EXI message and decode it on the fly.

EXI can utilize available schema information to improve compactness and performance, but does not depend on schemas to work. It works very effectively with partial schemas or in the absence of any schema. EXI provides several different encoding methods, including bit-aligned encoding, byte-aligned encoding, compressed encoding, uncompressed binary encoding, schema-informed encoding, "schema-less" encoding, etc.

Ref: http://www.w3.org/XML/EXI/

1.7.2.2 zlib

Zlib is a general-purpose compression algorithm that can produce an extremely compact message. It is free, relatively small and widely deployed on virtually every hardware platform. It has also been exhaustively tested over a period of years and is a proven technology.

Unlike EXI, using zlib would require to receive the entire message, decompress it, and then parse the resulting XML message. Also, zlib can be more processor intensive.

The comparative testing of EXI and zlib compression capabilities was done by Packetizer, Inc. The results of the test are shown in the figure below. It shows that although zlib almost always produces a smaller file, using EXI with the internal compression capability produces the most compact encoding of all.

Percentage of Original Size

File	Original XML	Schema	Schema Byte Aligned	Schema Lexical	No Schema	No Schema Byte Aligned	Precompression	Strict	Strict Byte Aligned	Strict Compressed	zlib
points.xml (126B)	100	40	54	40	40	53	54	40	54	44	71
notebook.xml (286B)	100	21	30	24	43	54	26	21	26	23	64
response-build.xml (1164B)	100	42	49	42	41	48	49	42	49	28	31
xtags-build.xml (2786B)	100	49	54	49	48	52	54	49	54	18	19
bsf-build.xml (8114B)	100	27	31	27	25	28	31	27	31	11	23
gnat-tags-build.xml (10KB)	100	27	31	27	25	28	31	27	31	12	24
periodic.xml (114KB)	100	8	14	13	17	21	11	8	11	4	7
factbook.xml (4MB)	100	54	57	54	54	57	55	54	55	16	24

Strict appears to provide roughly same performance as Schema ^^

Strict byte aligned appears to do as well as Precompression ^^

Compression gain for EXI and zlib (by Packetizer, Inc.)

Ref: http://www.zlib.net/

1.8 Other

1.8.1 GOES Image Product Distribution

Two systems are used to distribute imagery from the Geostationary Operational Environmental Satellites (GOES): the GOES Ingest and NOAAPORT Interface (GINI), and the GOES Low-Rate Information Transmission (LRIT).

1.8.1.1 GINI

The GOES Ingest and NOAAPORT Interface (GINI) system generates standard map projections from the GOES imager. The GINI crops, scales, and remaps this input into standard map projections such as Lambert Conformal, Mercator, and polar stereographic. GINI is the central remapping system for the NWS's Advanced Weather Interactive Processing System (AWIPS). The largest product created by GINI is the CONUS 1 km visible product (approx. 26 MB).

Remapped image files are sent to a NOAAPORT interface that will distribute images to more than 100 NWS weather forecast offices via a point-to-multipoint satellite broadcast. Up to 27 remapped sectors can be generated concurrently. The process of sector remapping is data driven; a predefined sector is generated when most of the data covering the specified geographic area is received.

The resulting output sectors are transmitted serially. Transmission of the highest priority sector begins immediately after the first data for the sector is received; lower priority sectors are then transmitted.

1.8.1.2 LRIT

The NOAA Low-Rate Information Transmission (LRIT) System became operational October 1, 2005. The LRIT processing system was developed and implemented by NOAA to provide a replacement for the Weather Facsimile (WEFAX) Processing System.

The digital Low Rate Information Transmission (LRIT) is an international standard for data transmission that was developed by the Coordination Group for Meteorological Satellites (CGMS) in response to a recommendation on digital meteorological satellite broadcasts. NOAA designed its LRIT system based on the CGMS standard. The NOAA LRIT system provides digital data, via a broadcast service, through its geostationary satellites.

Ref: http://www.nesdis.noaa.gov/

http://www.wmo.int/pages/prog/sat/CGMS/Directoryofapplications/en/ap9-31.htm

http://www.wmo.ch/pages/prog/sat/documents/CGMS-03_HRIT-LRIT_v2-6.pdf

http://www.nws.noaa.gov/noaaport/html/noaaport.shtml

1.8.2 Map Projection Formats

A map projection is a way to represent the Earth surface of a sphere on a plane. All map projections distort the surface in some fashion; therefore different map projections exist in order to preserve some properties of the sphere-like body at the expense of other properties. The major properties to operate with are area, shape, direction, bearing, distance, and scale.

A fundamental projection classification is based on the type of projection surface onto which the sphere is projected. The major surface types and, respectively, projection classes are:

- cylindrical
- conic
- azimuthal or plane

Some examples of the major projection classes are discussed in more details below.

1.8.2.1 Cylindrical Projections

1.8.2.1.1 Mercator

The Mercator projection has straight meridians and parallels that intersect at right angles. Scale is true at the equator or at two standard parallels equidistant from the equator. Meridians unequally spaced, distance increases away from equator directly proportional to increasing scale.

The projection was invented in 1569 by Gerardus Mercator (Flanders) graphically, and was standard for maritime mapping in the 17th and 18th centuries. In 19th and 20th centuries Mercator projection was widely used for mapping the world regions and oceans. Although the projection raised a lot of criticism recently, it is still popular for marine navigation because all straight lines on the map are lines of constant azimuth.

1.8.2.1.2 Transverse Mercator

Transverse Mercator projection results from projecting the sphere onto a cylinder tangent to a central meridian; distortions of scale, distance, direction and area increase away from the central meridian.

Transverse Mercator maps are often used to portray areas with larger north-south than east-west extent. Many national grid systems are based on the Transverse Mercator projection.

One specific use of the Transverse Mercator projection is the Universal Transverse Mercator (UTM) System. It is used to define horizontal positions world-wide by dividing the surface of the Earth into 6 degree zones, each mapped by the Transverse Mercator projection with a central meridian in the center of the zone. UTM zone numbers designate 6 degree longitudinal strips extending from 80 degrees South latitude to 84 degrees North latitude. UTM zone characters designate 8 degree zones extending north and south from the equator.

1.8.2.1.3 Plate Carre / Equidistant Cylindrical projection

Equidistant Cylindrical projection or Plate Carre is an unprojected coordinate system, which is formed by considering longitude and latitude as a simple rectangular coordinate system. Scale, distance, area, and shape are all distorted on these unprojected LAT/LON maps with the distortion increasing toward the poles. One distinct example of use of that projection is in Mission Control at NASA.

1.8.2.2 Conic Projections

1.8.2.2.1 Lambert Conformal

The standard projection for intermediate- and large-scale maps of regions in mid-latitudes, for which the transverse Mercator is not used, e.g. maps of North America. Area and shape are distorted away from standard parallels. Directions are true in limited areas.

1.8.2.2.2 Albers Equal Area Conic

It is a projection that distorts scale and distance except along standard parallels. Although Albers' projection is usually used with two standard parallels, it can be used with one.

Areas are proportional and directions are true in limited areas. Used in the United States and other large countries with a larger east-west than north-south extent.

1.8.2.3 Azimuthal Projections

1.8.2.3.1 Stereographic / Polar

The Stereographic Projection is constructed by projecting the globe onto a tangent plane from a "light source" located at the point on the globe antipodal to the plane's point of tangency. Stereographic projections are often used for navigation in Polar regions. Directions are true from the center point and scale increases away from the center point as well as distortion in area and shape.

1.8.2.3.2 Azimuthal Equidistant

Azimuthal equidistant projections are sometimes used to show air-route distances. Distances measured from the center are true. Distortion of other properties increases away from the center point.

Since the azimuthal equidistant projection shows every point on the globe at its correct distance, and in its correct direction, from the center of the projection, maps in this projection are very useful to people engaged in developing radio communication links.

1.8.2.3.3 Lambert Azimuthal Equal Area

The Lambert azimuthal equal-area projection is sometimes used to map large ocean areas. The central meridian is a straight line, others are curved. A straight line drawn through the center point is on a great circle.

Because it is azimuthal, it is well suited to mapping regions that do not have any large difference between their north-south extent and their east-west extent.

Ref: http://www.geo.hunter.cuny.edu/mp/